Newcastle University

COMPUTING SCIENCE

Generalising Traces

Ryszard Janicki, Jetty Kleijn, Maciej Koutny, and Lukasz Mikulski

TECHNICAL REPORT SERIES

No. CS-TR-1436 **October 2014**

TECHNICAL REPORT SERIES

No. CS-TR-1436 **October, 2014**

Generalising Traces

R. Janicki, J. Kleijn, M. Koutny, and L. Mikulski

Abstract

In the classical Mazurkiewicz trace approach the behaviour of a concurrent system is described in terms of sequential observations that differ only with respect to their ordering of independent actions. This paper aims to determine a full generalisation of the trace model to the case that actions can be observed as occurring simultaneously. Thus observations are sequences of steps, i.e., sets of actions. This leads to an extended trace model based on three relations between events: simultaneity, serialisability, and interleaving. Whereas the underlying causal structures of traces are based on dependencies between actions leading to a partial order interpretation, more general causal orders are needed to describe the invariant relations between the action occurrences in a generalised trace. We present a complete picture including extended dependence graphs and a characterisation in terms of the (most) general order structures.

Printed and published by Newcastle University,
Computing Science, Claremont Tower, Claremont Road,
Newcastle upon Tyne, NE1 7RU, England.

Bibliographical details

JANICKI, R., KLEIJN, J., KOUTNY, M., MIKULSKI, L.

Generalising Traces
[By] R. Janicki, J. Kleijn, M. Koutny, and L. Mikulski
Newcastle upon Tyne: Newcastle University: Computing Science, 2014.

(Newcastle University, Computing Science, Technical Report Series, No. CS-TR-1436)

Added entries

NEWCASTLE UNIVERSITY
Computing Science. Technical Report Series. CS-TR-1436

Abstract

In the classical Mazurkiewicz trace approach the behaviour of a concurrent system is described in terms of sequential observations that differ only with respect to their ordering of independent actions. This paper aims to determine a full generalisation of the trace model to the case that actions can be observed as occurring simultaneously. Thus observations are sequences of steps, i.e., sets of actions. This leads to an extended trace model based on three relations between events: simultaneity, serialisability, and interleaving. Whereas the underlying causal structures of traces are based on dependencies between actions leading to a partial order interpretation, more general causal orders are needed to describe the invariant relations between the action occurrences in a generalised trace. We present a complete picture including extended dependence graphs and a characterisation in terms of the (most) general order structures.

About the authors

Ryszard Janicki is a Professor at the Department of Computing and Software, McMaster University, Hamilton, Ontario, Canada.

Jetty Kleijn is a visiting fellow within the School of Computing Science, Newcastle University.

Maciej Koutny is a Professor in the School of Computing Science, Newcastle University. His research interests centre on the theory of distributed and concurrent systems, including both theoretical aspects of their semantics and application of formal techniques to the modelling and verification of such systems; in particular, model checking based on net unfoldings. He has also investigated non-interleaving semantics of priority systems, and the relationship between temporal logic and process algebras. Recently, he has been working on the development of a formal model combining Petri nets and process algebras as well as on Petri net based behavioural models of membrane systems.

Lukasz Mikulski obtained his PhD in 2012 from the University of Warsaw, Poland. He now works as a Lecturer at the Faculty of Mathematics and Computer Science of the Nicolaus Copernicus University, Torun, Poland.

Suggested keywords

TRACE
INDEPENDENCE
SIMULTANEITY
SERIALISABILITY
INTERLEAVING
EXTENDING CONCURRENCY ALPHABETS
TRACE OF STEP SEQUENCES
EXTENDED DEPENDENCE GRAPH
GENERALISED CAUSAL ORDER STRUCTURE

Generalising Traces

Ryszard Janicki[1], Jetty Kleijn[2], Maciej Koutny[3], and Łukasz Mikulski[3,4]

[1] Department of Computing and Software, McMaster University
Hamilton, ON, L8S 4K1, Canada
janicki@mcmaster.ca

[2] LIACS, Leiden University, P.O.Box 9512
NL-2300 RA Leiden, The Netherlands
h.c.m.kleijn@liacs.leidenuniv.nl

[3] School of Computing Science, Newcastle University
Newcastle upon Tyne NE1 7RU, United Kingdom
{maciej.koutny,lukasz.mikulski}@ncl.ac.uk

[4] Faculty of Mathematics and Computer Science, Nicolaus Copernicus University
Toruń, Chopina 12/18, Poland
lukasz.mikulski@mat.umk.pl

Abstract. In the classical Mazurkiewicz trace approach the behaviour of a concurrent system is described in terms of sequential observations that differ only with respect to their ordering of independent actions. This paper aims to determine a full generalisation of the trace model to the case that actions can be observed as occurring simultaneously. Thus observations are sequences of steps, i.e., sets of actions. This leads to an extended trace model based on three relations between events: simultaneity, serialisability, and interleaving. Whereas the underlying causal structures of traces are based on dependencies between actions leading to a partial order interpretation, more general causal orders are needed to describe the invariant relations between the action occurrences in a generalised trace. We present a complete picture including extended dependence graphs and a characterisation in terms of the (most) general order structures.

Keywords: trace, independence, simultaneity, serialisability, interleaving, extending concurrency alphabets, trace of step sequences, extended dependence graph, generalised causal order structure

1 Introduction

Mazurkiewicz traces [15, 16] are a well-established, classical, and basic model for representing and structuring sequential observations of concurrent behaviour; see, e.g., [1, 12].

The fundamental assumption underlying trace theory is that independent events (occurrences of actions) may be observed in any order. Sequences that differ only w.r.t. their ordering of independent events are identified as belonging to the same concurrent run of the system under consideration. Thus a trace is an equivalence class of sequences comprising all (sequential) observations of a single concurrent run. The dependencies between the events of a trace are invariant among (common to) all elements of the trace. This (acyclic) dependence graph determines through its transitive closure the underlying causality structure of the trace as a (labelled) partial order [19]. In fact this partial order can also be obtained as the intersection of the labelled total orders corresponding to the sequences forming the trace. Moreover, the linearisations (saturations) of this partial order correspond exactly to the sequences belonging to the trace. Thus a trace can be seen as a labelled partial order which is unique up to isomorphism, i.e., names of underlying elements; see, e.g., [1, 3, 12]. Moreover, the paper [20] provides the necessary connection (Szpilrajn's property) between the causal structures (partial orders) and observations (total orders), by showing that each partial order is the intersection of all its linearisation. The overall set-up can be summarised by the schematic commuting diagram shown in Figure 1.

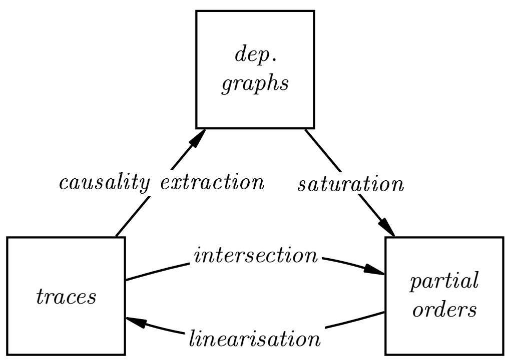


Fig. 1. Behaviour diagram for Mazurkiewicz traces.

Being based on equating independence and lack of ordering, the concurrency paradigm of Mazurkiewicz traces and the corresponding partial order interpretation of concurrency is rather restricted [6].

In this paper we carefully consider how to extend the trace approach to a more general situation by assuming that observers may not only register the occurrence of one action before another, but can also record simultaneous occurrences of actions. Thus here, observations consist of sequences of *steps*, i.e., sets of one or more actions that occur simultaneously. Still we aim at retaining the original philosophy underlying Mazurkiewicz traces and our set-up will be based on just a few explicit and simple design choices. Our considerations lead to the concept of a *fundamental concurrency alphabet* with three basic relations between pairs of different actions: *simultaneity* indicating that actions may occur together in a step; *serialisability* indicating a possible execution order for potentially simultaneous actions; and *interleaving* indicating that actions can *not* occur simultaneously though no specific ordering is required. These three relations can then be used to identify step sequences as observations of the same concurrent run. The resulting equivalence classes of step sequences are called *generalised traces*. It is the main aim of this paper to characterise such traces.

First however, we discuss concurrency alphabets in some more detail. The clear semantical meaning of the three relations — simultaneity, serialisability, interleaving — allows for an intuitive classification of some natural subclasses of fundamental concurrency alphabets. We present a hierarchy of interesting subcases among which the original concurrency alphabets of Mazurkiewicz and other types of concurrency alphabets known from the literature. Also two new non-trivial classes of concurrency alphabets are brought to light. Then a new, technically more convenient, definition of generalised traces is proposed on basis of concurrency alphabets with only two relations: simultaneity as before and *sequentialisability* which is a combination of serialisability and interleaving.

Next, we turn to the causal order structures underlying generalised traces with the ultimate aim to match generalised traces and step sequences with relational structures, just like Mazurkiewicz traces correspond to partial orders and total orders to sequences of action occurrences (see Figure 1). Partial orders are clearly not expressive enough to capture all possible relationships between events as determined by a generalised concurrency alphabet. Rather than a strict order (causality or 'before'), the relational structures we consider have a 'not later than' relation to represent weak causality (i.e., before or in the same step) and a 'mutual exclusion' relation for pure interleaving (not allowed in the same step but not causally ordered). Moreover, as shown in [6], weak causality and mutex are sufficient to represent the most general concurrent histories. We thus arrive at so-called *order structures*, labelled relational structures satisfying a separability (akin to acyclicity) and label-orderedness properties, as the counterpart of the dependence graphs underlying Mazurkiewicz traces. Step sequences correspond to saturated versions of these structures.

The order structures that satisfy a general variant of Szpilrajn's property (meaning that they can be obtained as the intersection of their saturated extensions) have been identified in [4] as general mutex order structures. Moreover, the closure of an order structure is a general mutex order structure. Thus we are left with the investigation of the properties of order structures obtained as (generalised) dependence graphs from step sequences. As expected, equivalent step sequences define the same dependence graph (order structure). It is however less obvious that, conversely, any step sequence (saturated order structure) derived from the dependence graph of a step sequence is equivalent with that step sequence (belongs to the same history). Eventually, the problem is reduced to the case of 'thin' step sequences in which every step is minimal in the sense that it cannot be split into a sequence of smaller steps, because its actions have to occur simultaneously. Interestingly, this leads to a proof technique similar to the approach for Mazurkiewicz traces consisting of sequences.

The whole discussion culminates in the development of a commutative diagram shown in Figure 5 for the most general model of traces based on step sequence observations, which is a counterpart of that the schematic diagram of Figure 1 that captures the relationship between traces and causal structures.

2 Mazurkiewicz traces

Mazurkiewicz traces stem from two elegant mathematical ideas which can be used to capture the essence of equivalence between different observations of the same run of a concurrent system. Both are based on a notion of independence between actions expressed as a binary relation ind. The first idea uses the concept of equations expressing partial commutativity of action occurrences as determined by the independence relation. As a result, sequences $wabu$ and $wbau$ of action occurrences are considered equivalent whenever $\langle a, b\rangle \in \mathsf{ind}$, irrespective of what w and u are. The second idea is the common partial order structure that underlies equivalent observations and is defined by the ordering of the occurrences of dependent actions. Thus, each trace, i.e., equivalence class of sequential observations, has a unique (upto isomorphism) labelled partial order as its signature.

Equations could, in general, be of the form $a_1 \ldots a_k = b_1 \ldots b_m$ where the a_i and b_j are actions with e.g., $c = de$ as a particular example. However, the usefulness for concurrency theory, of equations in this form is not obvious, unless there is an additional interpretation of the alphabet of actions which usually entails the need for operators. This, in particular, happens when, instead of sequences of actions, one considers sequences of sets of actions (or step sequences) together with the operation of set union.

The idea of considering equations on sets of actions generated by relations on actions has been used to define, e.g., comtraces [7, 14], g-comtraces [8], and interval traces [9]. Comtraces are a special case of absorbing monoids in the terminology of [8] — i.e., they are quotient monoids over step sequences derived from equations of the form $AB = A \uplus B$ — with the equations being derived from two relations, sim and ser, respectively called simultaneity and serialisability. Likewise, g-comtraces are a special case of partially commutative absorbing monoids in the terminology of [8] — i.e., they are quotient monoids derived from equations of the form $AB = A \uplus B$ and $AB = BA$ — with the equations being derived from simultaneity and serialisability as well as interleaving, inl. As shown in [13], the equations used in [8] and the subsequent papers do not model the relevant aspects of concurrent behaviours in a fully adequate way. The corresponding model of causal structures was also not fully satisfactory, and a suitable improvement was proposed in [4]. In essence, the problem was that the interleaving equations $AB = BA$ were defined only by $A \times B \subseteq \mathsf{inl}$, in effect disallowing the mixing of two different 'reasons' for commuting two actions; the other one being $A \times B \subseteq \mathsf{ser} \cap \mathsf{ser}^{-1}$ (for detailed discussion see Section 4).

In this paper, we will take a fresh look at the way in which a theory of traces consisting of step sequences could be developed and, in particular, we will develop an improved treatment of equations on step sequences of [8]. The soundness of the proposed improvement will be demonstrated in the second part of the paper by showing how a recently proposed model of causal structures matches exactly the extension of Mazurkiewicz traces introduced here.

3 Preliminaries

We use standard notions of set theory and formal language theory.

Alphabets, sequences and step sequences. Throughout the paper,

$$\boxed{\Sigma \text{ is an } \textit{alphabet} \text{ of actions}} \quad \text{and} \quad \boxed{\mathbb{S} \text{ is the set of all } \textit{steps} \text{ over } \Sigma}$$

An alphabet is assumed to be a finite and nonempty set, and steps are nonempty subsets of an alphabet of actions.

We use SEQ to denote the set Σ^* consisting of all finite sequences of elements from Σ (*sequences* over Σ), and SSEQ to denote the set $\mathbb{S}^*$ consisting of all finite sequences of steps over Σ (*step sequences* over Σ). Moreover, λ denotes both the empty sequence and the empty step sequence.

We will often identify a singleton step $\{a\}$ with its only member, a, tacitly assuming that $\Sigma \subset \mathbb{S}$. Moreover, we will denote non-singleton steps by listing their elements within parentheses. Thus a step sequence $\{a\}\{b,c\}\{a\}$ can be written down as $a(bc)a$ or $a(cb)a$.

Let $u = A_1 \dots A_k \in \mathbb{S}^*$ be a step sequence. Then:

- for every action $a \in \Sigma$, $\#_u(a)$ is the number of occurrences of a within u;
- $occ(u) = \{\langle a, i\rangle \mid a \in \Sigma \wedge 1 \le i \le \#_u(a)\}$ is the set of *action occurrences* of u;
- the *position* $pos_u(\alpha)$ within u of an action occurrence $\alpha = \langle a, i\rangle \in occ(u)$ is the smallest index $j \le k$ such that the number of occurrences of a within $A_1 \dots A_j$ is exactly i;
- for every $i \le k$, $occ(u,i) = \{\alpha \in occ(u) \mid pos_u(\alpha) = i\}$ are the action occurrences contributing to the i-th step; and
- $occseq(u) = occ(u,1) \dots occ(u,k)$ is u with explicitly listed action occurrences.

For example, $occ(a(bc)a) = \{\langle a,1\rangle, \langle a,2\rangle, \langle b,1\rangle, \langle c,1\rangle\}$, $pos_{a(bc)a}(\langle a,2\rangle) = 3$, and $occseq(u) = \langle a,1\rangle\{\langle b,1\rangle, \langle c,1\rangle\}\langle a,2\rangle$.

Functions. A mapping $f : X \to Y$ is denoted by $X \xrightarrow{f} Y$, and we use $x \xmapsto{f} y$ to denote $f(x) = y$. Moreover, if $X' \subseteq X$ and $Y' \subseteq Y$ are such that $f(X') \subseteq Y'$, the restriction of f to the domain X' and codomain Y' is denoted by $X' \xrightarrow{f} Y'$. Finally, for $R \subseteq X \times X$ we let $f(R) = \{(f(x), f(y)) \mid (x,y) \in R\}$.

Relations. $id_X = \{\langle x, x\rangle \mid x \in X\}$ is the identity relation on a set X, and $R \circ Q = \{\langle w, z\rangle \mid \exists x \in X :\; wRx \wedge xQz\}$ is the composition of two binary relations, R and Q, over X. Moreover, if $P \subseteq X \times X$, then we define

$$R \circ_P Q = \{\langle w, z\rangle \mid \exists \langle x, y\rangle \in P :\; wRxQz \;\wedge\; wRyQz\}\ .$$

Remark 1. $R \circ_P Q$ can be thought of as a composition of R and Q *supported* by P, as can be seen in the diagram below which illustrates the derivation of $\langle w, z\rangle \in R \circ_P Q$:

$$\begin{tikzcd} & x \arrow[d] \arrow[dr, dotted] & \\ w \arrow[ur, dashed] \arrow[dr, dashed] & y \arrow[r, phantom, ""] & z \\ & y \arrow[ur, dotted] & \end{tikzcd}$$

$R : \dashrightarrow \qquad Q : \cdots\!\!\rightarrow \qquad P : \longrightarrow$

Note that $\circ = \circ_{id_X}$ and $R \circ_P Q \subseteq R \circ Q$. $\diamond$

The inverse of a binary relation R is given by $R^{-1} = \{\langle y, x\rangle \mid \langle x, y\rangle \in R\}$, and the symmetric closure by $R^{sym} = R \cup R^{-1}$. Moreover, a relation $R \subseteq X \times X$ is:

– symmetric if $R = R^{-1}$, and asymmetric if $R \cap R^{-1} = \varnothing$;
– reflexive if $id_X \subseteq R$, and irreflexive if $id_X \cap R = \varnothing$;
– transitive if $R \circ R \subseteq R$;
– an equivalence relation if it is symmetric, transitive and reflexive;
– a partial order relation if it is irreflexive and transitive;
– a weak partial order relation if it is reflexive and $R \setminus id_X$ is a partial order relation;
– a total order relation if it is a partial order relation such that we have $R^{sym} = (X \times X) \setminus id_X$; and
– a stratified partial order relation if it is a partial order relation such that $(X \times X) \setminus R^{sym}$ is an equivalence relation (note that each total order relation is also a stratified partial order relation).

Given $R \subseteq X \times X$, $R^0 = id_X$ and $R^n = R^{n-1} \circ R$, for all $n \geq 1$. Then:

– $R \cup id_X$ is the reflexive closure of R;
– $R^+ = \bigcup_{i \geq 1} R^i$ is the transitive closure of R;
– $R^* = \bigcup_{i \geq 0} R^i = R^+ \cup id_X$ is the reflexive transitive closure of R;
– $R^{\curlywedge} = R^+ \setminus id_X = R^* \setminus id_X$ is the irreflexive transitive closure of R;
– $R^{\circledast} = R^* \cap (R^*)^{-1} = (R^{\curlywedge} \cap (R^{\curlywedge})^{-1}) \uplus id_X$ is the largest equivalence relation contained in R^*; and
– R is acyclic if R^+ is asymmetric.

A *labelled partial order* is a triple $po = \langle \Delta, \prec, \ell\rangle$, where Δ, the domain of po, is a finite set, $\Delta \xrightarrow{\ell} \Sigma$ is a labelling of the domain elements, and $\prec$ is an irreflexive transitive binary relation on Δ. We will use Δ_{po}, $\prec_{po}$, and ℓ_{po} to denote the respective components of po. A labelled partial order po is:

– *total* or *stratified* if the partial order relation $\prec_{po}$ is respectively total or stratified;
– *label-ordered* if $x \prec_{po}^{sym} y$, for all distinct $x, y \in \Delta_{po}$ satisfying $\ell_{po}(x) = \ell_{po}(y)$ (hence all the elements with the same label are totally ordered by $\prec_{po}$).

A *linearisation* of an acyclic binary relation $\ll$ over a finite set X is any enumeration $u = x_1 \dots x_k$ of the elements of X such that $x_i \ll x_j$ implies $i < j$, for all $i, j \leq k$. Furthermore, we write $u \bowtie w$ if $w = x_1 \dots x_{i-1} x_{i+1} x_i x_{i+2} \dots x_k$, where x_i and x_{i+1} are such that $x_i \not\ll x_{i+1} \not\ll x_i$. It turns out that the linearisations of $\ll$ can be related via repeated applications of $\bowtie$.

Proposition 1. *If u and w are linearisations of an acyclic relation $\ll$ over a finite set X, then $u \bowtie^* w$.*

Proof. We proceed by induction on $|X|$. In the base case, $|X| = 0$, we have that both u and v are the empty enumeration. In the inductive case, $|X| > 0$, we proceed as follows.

Since X is nonempty and finite, and $\ll$ acyclic, there is an $x \in X$ such that there is no $y \in X' = X \setminus \{x\}$ such that $y \ll x$. We now observe that there is a u' such that $u \bowtie^* xu'$. Indeed, suppose that $u = y_1 \dots y_m x u''$. Then, for all $i \leq m$, we have $y_i \not\ll x$ (by the choice of x), and $x \not\ll y_i$ (by u being a linearisation of $\ll$). Hence $u \bowtie^* x y_1 \dots y_m u''$. Similarly, there is w' such that $w \bowtie^* xw'$. We now observe that u' and w' are linearisations of $\ll' = \ll \cap (X' \times X')$. Hence, by the induction hypothesis, $u' \bowtie^* w'$. As a consequence, $xu' \bowtie^* xw'$ and so $u \bowtie^* xu' \bowtie^* xw' \bowtie^* w$. $\square$

Equations on step sequences. Let EQ be a finite set of equations on step sequences, each equation being of the form $u = v$, where u and v are nonempty step sequences. This set of equations induces a relation $\approx_{EQ}$ on step sequences comprising all pairs $\langle tuw, tvw \rangle$ such that $t, w \in \mathbb{S}^*$, and $u = v$ or $v = u$ is an equation in EQ. Furthermore, $\equiv_{EQ}$ is the equivalence relation on step sequences defined as $\approx^*_{EQ}$.

Relational structures. To represent observational as well as causal relationships between action occurrences, we will use *relational structures*

$$\boxed{rs = \langle \Delta, \rightleftharpoons, \sqsubset, \ell \rangle}$$

comprising a finite *domain*, Δ, two binary relations $\rightleftharpoons$ and $\sqsubset$ on Δ, and a domain labelling, $\Delta \xrightarrow{\ell} \Sigma$. Moreover, we will use $\prec$ to denote the intersection of $\sqsubset$ and $\rightleftharpoons$, and $\prec^a$ to denote $\prec$ restricted to $\ell^{-1}(a)$, for every label $a \in \Sigma$. We will often write Δ_{rs} $\rightleftharpoons_{rs}$, $\sqsubset_{rs}$, ℓ_{rs}, $\prec_{rs}$ and $\prec^a_{rs}$ to emphasize the relational structure rs.

Two relational structures, rs and rs', are *isomorphic* if there exists a bijection $\Delta_{rs} \xrightarrow{\kappa} \Delta_{rs'}$ such that $rs' = \langle \Delta_{rs'}, \kappa(\rightleftharpoons_{rs}), \kappa(\sqsubset_{rs}), \ell_{rs} \circ \kappa^{-1} \rangle$. We denote this by $rs \sim_\kappa rs'$ or $rs \sim rs'$.

There are two notions which will play a key role in our treatment of relational structures. The first notion allows one to make the structure more specific, by adding to the two component relations.

A relational structure rs' is an *extension* of a relational structure rs with the same domain and labelling, if $\rightleftharpoons_{rs}$ and $\sqsubset_{rs}$ are respectively included in $\rightleftharpoons_{rs'}$ and $\sqsubset_{rs'}$. We denote this by $rs' \in \mathsf{ext}(rs)$ or $rs \lhd rs'$.

The second notion is *structure-closure*, similar to the transitive closure of an acyclic relation R, which yields the least partial order relation which includes R. Also here the closure is defined in terms of two families of relational structures, one being a subset of the other. Structures belonging to the smaller family are closed and the closure of a structure belonging to the larger family is obtained by extending its relations leading to a structure in the other family.

Let $\mathsf{F} \supset \mathsf{F}'$ be two families of relational structures. A *structure-closure operator of* F *with respect to* F' is a mapping $\mathsf{F} \xrightarrow{\mathsf{cls}} \mathsf{F}'$ such that, for every $rs \in \mathsf{F}$:

$$rs \lhd \mathsf{cls}(rs) \tag{1}$$

and, for all $rs \in \mathsf{F}$ and $rs' \in \mathsf{F}'$:

$$rs \lhd rs' \implies \mathsf{cls}(rs) \lhd rs' \ . \tag{2}$$

We then obtain that closing a closed structure has no effect, and all the closed extensions of a relational structure are also extensions of the closure of that structure, i.e., closing a structure does not enlarge 'too much' the component relations.

Proposition 2. *Let* $\mathsf{F} \xrightarrow{\mathsf{cls}} \mathsf{F}'$ *be a structure-closure operator. Then, for all* $rs \in \mathsf{F}$ *and* $rs' \in \mathsf{F}'$*:*

1. $\mathsf{cls}(rs') = rs'$.
2. $\mathsf{ext}(rs) \cap \mathsf{F}' = \mathsf{ext}(\mathsf{cls}(rs)) \cap \mathsf{F}'$. $\diamond$

Proof. ***(1)*** By Eq.(1), $rs' \lhd \mathsf{cls}(rs')$. Moreover, $rs' \lhd rs'$ and so, by Eq.(2), $\mathsf{cls}(rs') \lhd rs'$. Hence $rs' \lhd \mathsf{cls}(rs') \lhd rs'$, and so $\mathsf{cls}(rs') = rs'$.

(2) Let $rs'' \in \mathsf{F}'$. We need to show that $rs \lhd rs''$ iff $\mathsf{cls}(rs) \lhd rs''$. The left-to-right implication follows from Eq.(2). Moreover, the right-to-left implication follows from Eq.(1). $\square$

Three properties relevant to the relations between action occurrences are defined as follows. A relational structure $rs = \langle \Delta, \rightleftharpoons, \sqsubset, \ell \rangle$ is:

- *separable* if $\rightleftharpoons$ is symmetric, $\sqsubset$ is irreflexive, and $\rightleftharpoons \cap \sqsubset^{\circledast} = \varnothing$ (note that this implies that $\rightleftharpoons$ is also irreflexive as id_Δ is included in $\sqsubset^{\circledast}$);
- *label-ordered* if $x \prec y$ or $y \prec x$, for all $x \neq y$ satisfying $\ell(x) = \ell(y)$; and
- *label-linear* if $\prec^a$ is a total order relation, for every label $a \in \Sigma$.

If a relational structure is label-ordered it is guaranteed that domain elements with the same label (intuitively representing two occurrences of the same action) are related by $\prec$. In combination with separability it is a partial order as we prove next.

Proposition 3. *Every separable label-ordered relational structure is label-linear.*

Proof. Let $rs = \langle \Delta, \rightleftharpoons, \sqsubset, \ell \rangle$ be a separable label-ordered relational structure. Suppose that $a \in \Sigma$ and $x, y, z \in \ell^{-1}(a)$ and $x \prec z \prec y$. First, we observe that $x \neq y$ since otherwise we would obtain a contradiction with the separability of rs. Hence by rs being label-ordered, we have $x \prec^{sym} y$. If $y \prec x$, we again obtain a contradiction with the separability of rs. Hence $x \prec y$. □

The intuitive meaning of separability and label-linearity in the context of concurrent systems behaviour will be discussed in Section 5.1. Here we focus on technical properties. Isomorphisms between label-linear relational structures are unique.

Proposition 4. *If there is a bijection establishing an isomorphism between two label-linear relational structures, then it is unique.*

Proof. Let $rs \sim_\kappa rs'$ be isomorphic label-linear relational structures, and let $a \in \Sigma$. By the label-preservation of κ, κ is a bijection between $\ell_{rs}^{-1}(a)$ and $\ell_{rs'}^{-1}(a)$. Hence, by the label-linearity of rs, κ restricted to $\ell_{rs}^{-1}(a)$ is unique. □

A nonempty set rss of relational structures is *consistent* if all these relational structures have the same domain Δ and domain labelling ℓ. For such a set, the *intersection* is the relational structure:

$$\bigcap rss = \langle \Delta, \bigcap_{rs \in rss} \rightleftharpoons_{rs}, \bigcap_{rs \in rss} \sqsubset_{rs}, \ell \rangle \; .$$

A consistent rss is said to be *separable* or *label-ordered* or *label-linear* if so is the intersection $\bigcap rss$.

Proposition 5. *Let rss be a consistent set of relational structures.*

1. *If rss is label-ordered, then so are all its elements.*
2. *If at least one element of rss is separable, then so is rss.*

Proof. Follows directly from the definitions. □

Note that the implications in the above proposition cannot be reversed. Moreover, it is not the case that the relational structures belonging to a label-linear rss have to be label-linear.

In this paper, we will be interested in sets of label-linear relational structures.

Proposition 6. *Let rss be a label-linear consistent set of label-linear relational structures, and $a \in \Sigma$ be a label. Then $\prec^a_{\bigcap rss} = \prec^a_{rs}$, for all $rs \in rss$.*

Proof. Clearly, $\prec^a_{\bigcap rss} \subseteq \prec^a_{rs}$. Moreover, $\prec^a_{rs} \subseteq \prec^a_{\bigcap rss}$ as otherwise $\prec^a_{\bigcap rss}$ would not be a total order relation (note that $\prec^a_{rs}$ is a total order relation). □

Two label-linear consistent sets of label-linear relational structures, rss and rss', are *isomorphic* if there are bijections $\Delta_{rss} \xrightarrow{\kappa} \Delta_{rss'}$ and $rss \xrightarrow{\phi} rss'$ such that $rs \sim_\kappa \phi(rs)$, for all $rs \in rss$. We denote this by $rss \sim rss'$ or $rss \sim_{\kappa,\phi} rss'$.

Proposition 7. *Two label-linear consistent sets rss and rss′ of label-linear relational structures are isomorphic if and only if for each relational structure in one set there is an isomorphic relational structure in the other set.*

Proof. ($\Longrightarrow$) Follows from the definition of isomorphism between rss and rss'.

($\Longleftarrow$) First, we observe that all relational structures within rss (and also within rss') are non-isomorphic. Indeed, suppose that $rs \sim_\kappa rs'$, for some relational structures $rs, rs' \in rss$. Then, by Proposition 6, we have that κ is the identity on Δ_{rss}. Hence $rs = rs'$. It therefore follows that there is a unique bijection $rss \xrightarrow{\phi} rss'$ relating isomorphic relational structures.

Suppose now that $rs \sim_\kappa \phi(rs)$ and $rs' \sim_{\kappa'} \phi(rs')$. By Proposition 4, both κ and κ' are unique isomorphisms. It then follows from Proposition 6 that $\kappa|_{\ell^{-1}(a)} = \kappa'|_{\ell^{-1}(a)}$, for every $a \in \Sigma$. Hence $\kappa = \kappa'$. □

Finally, we obtain the uniqueness of isomorphisms between label-linear consistent sets of label-linear relational structures.

Proposition 8. *If there are bijections (κ, ϕ) establishing an isomorphism between two label-linear consistent sets rss and rss′ of label-linear relational structures, then each of them is unique.*

Proof. Using similar arguments as in the proof of Proposition 7. □

In conclusion, label-linearity is a powerful notion which essentially allows one to completely abstract from the identities of the underlying domain elements.

4 Extending Mazurkiewicz traces

Originally, Mazurkiewicz trace theory is concerned with adding structure to the otherwise plain set of observations of the behaviour of a concurrent system represented by sequences of action occurrences. Action occurrences are atomic and it is assumed that there is a (static) notion of independence between pairs of actions. This independence relation is then used to identify observations which differ only by the order of occurrences of independent actions. The resulting equivalence relation groups together observations of the same concurrent run (history), and the corresponding equivalences classes are called *traces*. The importance of the resulting model is reinforced by the fact that it corresponds to an order theoretic model of partial order histories of concurrent systems and concurrent system models.

In this paper, we aim at a full generalisation of the theory of Mazurkiewicz traces to the case that the smallest unit of observation is a set of actions (a step) rather than a single action, reflecting the idea that actions could occur (and be observed as occurring) simultaneously. Thus behavioural observations are now represented by step sequences rather than sequences of action occurrences. We will now formulate our generalisation in stages, trying to retain the philosophy behind the original model, and to make all design choices and decisions both explicit and lightweight.

The first question we face is what should be the form of the equations used in the extended model. One type, *interleaving* equations of the form

$$\boxed{AB = BA}$$

is a direct lifting of Mazurkiewicz's interleaving equations, $ab = ba$, to the domain of step sequences. However, restricting ourselves to only interleaving equations would effectively mean that the resulting traces of step sequences would be Mazurkiewicz traces of sequences, with each element of such a sequence being a step. However, no full generality is then obtained since we would not be able to derive $(ab) = ab$ if actions a and b were observed as simultaneously but could also have been observed as a followed by b. We will therefore use a second form of equations

$$\boxed{C = DE}$$

intended to capture the *serialisation* of a step C into two consecutive substeps, D and E. In a way, this is the lightest application of the inherent structuring of sets through subset inclusion. *No other equations will be considered in this paper.* In our discussion we will assume that

$$\boxed{A \cap B = \varnothing}$$

reflecting the underlying philosophy that the order of occurrences of the same action should not be changed. Similarly, we will assume that in the above

$$\boxed{D \cap E = \varnothing}$$

reflecting the commonly held view that in equivalent observations each action should occur the same number of times.

Having decided on the form of equations, $AB = BA$ and $C = DE$, it is definitely not our intention to consider all such equations. Rather, we are interested in equations which can be derived using the fundamental principle of Mazurkiewicz's approach that *all relevant equivalences between behaviours are derived from binary relationships between actions.* In our case, this amounts, in particular, to deciding what steps can be commuted, similarly as in the original approach, and which steps can be split and how. Moreover, we need to specify which sets of actions are legal steps (i.e., can occur simultaneously). We will address these issues by assuming that the equations are derived from three fundamental relationships:

$\boxed{\mathsf{sim} \subseteq \Sigma \times \Sigma}$

is the *simultaneity* relation defining all legal steps $A \in \mathbb{S}$ through the requirement that $(A \times A) \setminus id_\Sigma \subseteq \mathsf{sim}$. We further stipulate that sim is irreflexive and symmetric. The former assumption is not necessary from a technical point of view, but is added explicitly to reflect the idea that actions do not occur simultaneously with themselves. The second assumption simply reflects the fact that in realistic observations of system behaviour simultaneity is a symmetric relationship.

$\boxed{\mathsf{inl} \subseteq \Sigma \times \Sigma}$

defining the *interleaving* equations $AB = BA$ through $A \times B \subseteq \mathsf{inl}$, with A and B being legal steps. We further stipulate that inl is irreflexive, symmetric, and $\mathsf{inl} \cap \mathsf{sim} = \varnothing$. Irreflexivity and symmetry correspond to the properties of Mazurkiewicz's independence relation (and guarantee that $A \cap B = \varnothing$). $\mathsf{inl} \cap \mathsf{sim} = \varnothing$ means that at this point we interleave only events which cannot occur simultaneously (but see the discussion below).

$\boxed{\mathsf{ser} \subseteq \Sigma \times \Sigma}$

defining the *serialisability* equations $C = DE$ through $D \times E \subseteq \mathsf{ser}$ and $C = D \cup E$, with C, D, E being legal steps. We further stipulate that ser is irreflexive and $\mathsf{ser} \subseteq \mathsf{sim}$. Irreflexivity implies that no event can be serialised with itself and guarantees that $D \cap E = \varnothing$. $\mathsf{ser} \subseteq \mathsf{sim}$ means that at this point we serialise only events which can occur simultaneously (but see the discussion below).

In the final stage of the construction of the extended model of traces, let us assume that we have two steps, A and B, such that $A \times B \subseteq \mathsf{ser} \cap \mathsf{ser}^{-1}$. In that case we have two equations, $A \cup B = AB$ and $B \cup A = BA$, and hence a derived equivalence $AB \equiv BA$. Intuitively, for each pair $\langle a, b\rangle \in A \times B$ there is a reason to commute. Taking this observation further, also in the case of two steps, A' and B', such that for each pair $\langle a, b\rangle \in A' \times B'$ either $\langle a, b\rangle \in \mathsf{inl}$ or $\langle a, b\rangle \in \mathsf{ser} \cap \mathsf{ser}^{-1}$, there is sufficient reason to commute A' and B'; in other words, $A'B' \equiv B'A'$. We will therefore require that the interleaving equations $AB = BA$ are defined through $A \times B \subseteq \mathsf{inl} \cup (\mathsf{ser} \cap \mathsf{ser}^{-1})$ rather than by $A \times B \subseteq \mathsf{inl}$.
This basically concludes the design of our extended trace model, and leads to the following formalisation of our extension of the trace model first introduced by Mazurkiewicz.

4.1 Fundamental concurrency alphabets

The three relations sim, inl, and ser described above are the basic building blocks of the new, extended concurrency alphabets. As such they will define a set of equations and then an equivalence relation for step sequences over Σ.

Definition 1 (fundamental concurrency alphabet). *A* fundamental concurrency alphabet *is a quadruple*

$$\boxed{\psi = \langle \Sigma, \mathsf{sim}, \mathsf{inl}, \mathsf{ser}\rangle}$$

where $\mathsf{sim}, \mathsf{inl}, \mathsf{ser}$ *are irreflexive relations over* Σ *such that* sim *and* inl *are symmetric,* $\mathsf{inl} \cap \mathsf{sim} = \varnothing$ *and* $\mathsf{ser} \subseteq \mathsf{sim}$*. The family of all fundamental concurrency alphabets will be denoted by* Ψ*.* ◇

The set of *steps* defined by a fundamental concurrency alphabet ψ is given by:

$$\boxed{\mathbb{S}_\psi = \{A \subseteq \Sigma \mid A \neq \varnothing \wedge (A \times A) \setminus id_\Sigma \subseteq \mathsf{sim}\}}$$

and the *equations* EQ_ψ induced by ψ are as follows, where $A, B \in \mathbb{S}_\psi$:

$$\boxed{\begin{array}{lll} AB =_\psi BA & \text{if } A \times B \subseteq \mathsf{inl} \cup (\mathsf{ser} \cap \mathsf{ser}^{-1}) & (\textit{interleaving}) \\ AB =_\psi A \cup B & \text{if } A \times B \subseteq \mathsf{ser} & (\textit{serialisability}) \end{array}}$$

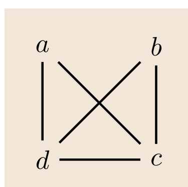
The resulting relations $\approx_{EQ_\psi}$ and $\equiv_{EQ_\psi}$ on step sequences will respectively be denoted by $\approx_\psi$ and $\equiv_\psi$, and the set of equivalence classes of $\equiv_\psi$ which contain at least one step sequence in $\mathsf{SSEQ}_\psi = \mathbb{S}_\psi^*$, called *traces* of step sequences, by TSSEQ_ψ. Moreover, the trace containing $u \in \mathsf{SSEQ}_\psi$ will be denoted by $[\![u]\!]_\psi$.

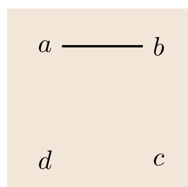
Applying the equations in EQ_ψ to step sequences composed of legal steps can never produce an illegal step.

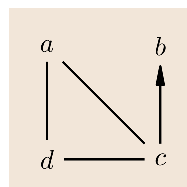
Proposition 9. *If* $\tau \in \mathsf{TSSEQ}_\psi$ *then* $\tau \subseteq \mathsf{SSEQ}_\psi$.

Proof. Follows from the fact that $\mathsf{ser} \subseteq \mathsf{sim}$, and so if we consider an equation $AB =_\psi A \cup B$ with $A, B \in \mathbb{S}_\psi$, we have that $A \cup B \in \mathbb{S}_\psi$. □

Example 1. Consider $\psi_0 = \langle \{a, b, c, d\}, \mathsf{sim}, \mathsf{inl}, \mathsf{ser} \rangle$, a fundamental concurrency alphabet with simultaneity, interleaving, and serialisability relations given below, where each edge stands for two arrows in opposite directions:

$\mathsf{sim} =$ 

$\mathsf{inl} =$ 

$\mathsf{ser} =$ 

ψ_0 generates, e.g., the interleaving equations $ab =_{\psi_0} ba$ and $a(bd) =_{\psi_0} (bd)a$, and serialisability equations $(ac) =_{\psi_0} ac$, $(ac) =_{\psi_0} ca$, and $(bc) =_{\psi_0} cb$. We also have:

$$\begin{array}{ll}
[\![(bd)c]\!]_{\psi_0} & = \{(bd)c\} \\
[\![c(bd)]\!]_{\psi_0} & = \{c(bd), (cbd)\} \\
[\![a(bd)]\!]_{\psi_0} & = \{a(bd), (bd)a\} \\
[\![a(bc)]\!]_{\psi_0} & = \{a(bc), (bc)a, acb, cba, cab, (ca)b\} \\
[\![a(cd)]\!]_{\psi_0} & = \{(acd), a(cd), (ac)d, (ad)c, c(ad), (cd)a, d(ac), acd, adc, cad, cda, \\
& \qquad dac, dca\} \\
[\![a(bcd)]\!]_{\psi_0} & = \{a(bcd), (bcd)a, ac(bd), ca(bd), c(bd)a, (ac)(bd)\}
\end{array}$$

We also note that $(bc) =_{\psi_0} bc$ is not an equation generated by ψ_0. ◇

4.2 Classifying fundamental concurrency alphabets

Although the main aim of this paper is to characterise generalised traces generated by fundamental concurrency alphabets, it is worth looking at some restricted classes of alphabets, especially in view of the fact that the three relations involved in such alphabets have a clear semantical meaning. For example,

$\mathsf{sim} \setminus \mathsf{ser} = \varnothing$ means that each step can be split into sequences in every possible way. Note that to be able to split a step into at least one sequences it is enough to require acyclicity of the relation $\mathsf{sim} \setminus \mathsf{ser}$ [17] (the resulting model being close to Vogler's ST-traces [21]), whereas $\mathsf{ser} = \varnothing$ means that there are no serialisability equations at all.

To obtain a classification of fundamental concurrency alphabets, we recall that $\mathsf{ser} \subseteq \mathsf{sim}$ and $\mathsf{sim} \cap \mathsf{inl} = \varnothing$, and then observe that the following are the three distinct components in the Venn diagram of the relations sim, inl and ser:

$$\boxed{\mathsf{sim} \setminus \mathsf{ser} \qquad \mathsf{inl} \qquad \mathsf{ser}}$$

Hence, in a rather natural way, we can distinguish eight classes of fundamental concurrency alphabets, as shown in Figure 2, where the subscripts indicate which relations are empty. Thus, for example, $\Psi_{\mathsf{inl} \cup \mathsf{ser}}$ comprises all fundamental concurrency alphabets such that $\mathsf{inl} \cup \mathsf{ser} = \varnothing$.

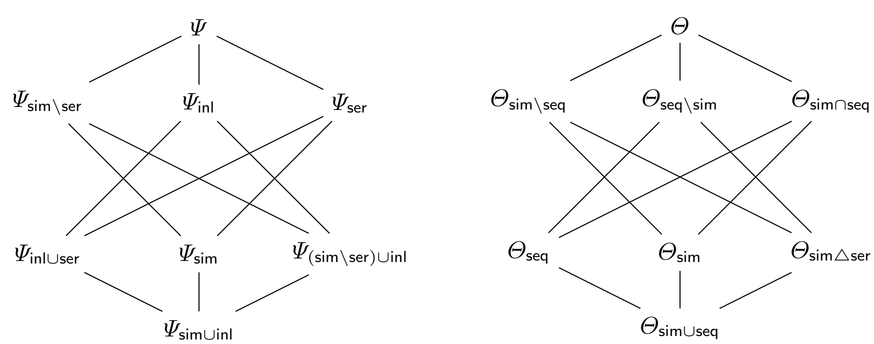


Fig. 2. The lattice of eight types of fundamental concurrency alphabets (left), and the corresponding eight types of generalised concurrency alphabets (right). A linked model depicted higher is more general. Two classes of corresponding alphabets occupy the same positions in the two diagrams (see Theorem 2). Note: $\mathsf{sim} \triangle \mathsf{ser} = (\mathsf{sim} \setminus \mathsf{seq}) \cup (\mathsf{seq} \setminus \mathsf{sim})$ denotes the symmetric difference of sim and ser.

We now briefly look at the eight classes of fundamental concurrency alphabets:

$\boxed{\Psi}$

is the family of all fundamental concurrency alphabets.

$\boxed{\Psi_{\mathsf{sim} \setminus \mathsf{ser}}}$

comprises alphabets such that the serialisability equations are rich enough to split any step in every possible way.

$\boxed{\Psi_{\mathsf{inl}}}$

comprises alphabets without true interleaving, and so one does not need interleaving equations. Note, however, that steps can be split. In the literature, alphabets in Ψ_{inl} are called *comtrace alphabets* [6], after dropping the empty interleaving relation inl.

$\boxed{\Psi_{\mathsf{ser}}}$

comprises alphabets where the only manipulation on steps is done through interleaving equations.

$\boxed{\Psi_{\mathsf{inl}\cup\mathsf{ser}}}$

comprises alphabets which generate traces comprising just one step sequence.

$\boxed{\Psi_{\mathsf{sim}}}$

comprises alphabets which do not involve true step sequences, and so one does not not need serialisability equations.

$\boxed{\Psi_{(\mathsf{sim}\setminus\mathsf{ser})\cup\mathsf{inl}}}$

comprises alphabets without interleaving equations, but the serialisability equations are rich enough to split and reorder steps in every possible way. Since $\mathsf{sim} = \mathsf{ser}$ and $\mathsf{inl} = \varnothing$, the serialisability ser and interleaving inl relations are dropped.

$\boxed{\Psi_{\mathsf{sim}\cup\mathsf{inl}}}$

comprises alphabets which generate traces comprising just one sequence.

To summarise, the alphabets in $\Psi_{\mathsf{sim}\cup\mathsf{inl}}$ and $\Psi_{\mathsf{inl}\cup\mathsf{ser}}$ are of little interest; the alphabets in Ψ_{inl}, Ψ_{sim} and $\Psi_{(\mathsf{sim}\setminus\mathsf{ser})\cup\mathsf{inl}}$ have been considered in the literature; and those in Ψ will be discussed in this paper. This leaves $\Psi_{\mathsf{sim}\setminus\mathsf{ser}}$ and Ψ_{ser} as new non-trivial types of concurrency alphabets which are worthy of further investigation.

4.3 An alternative presentation of fundamental concurrency alphabets

The above direct capture of a concurrency alphabet can be replaced by a simpler notion, based on two rather than three relations on actions: one is sim as above defining all legal steps, whereas the other one, seq, combines serialisability and (pure) interleaving.

Definition 2 (generalised concurrency alphabet). *A* generalised concurrency alphabet *is a triple*

$$\boxed{\theta = \langle \Sigma, \mathsf{sim}, \mathsf{seq} \rangle}$$

where sim *and* seq *(called* sequentialisability*) are irreflexive relations over Σ such that* sim *and* $\mathsf{seq} \setminus \mathsf{sim}$ *are symmetric. The family of all generalised concurrency alphabets will be denoted by Θ.* ◇

The set of *steps* defined by a generalised concurrency alphabet θ is given by:

$$\boxed{\mathbb{S}_\theta = \{A \subseteq \Sigma \mid A \neq \varnothing \wedge (A \times A) \setminus id_\Sigma \subseteq \mathsf{sim}\}}$$

and the *equations* EQ_θ induced by θ are as follows, where $A, B \in \mathbb{S}_\theta$:

$$\boxed{\begin{array}{lll} AB =_\theta BA & \text{if } A \times B \subseteq \mathsf{seq} \cap \mathsf{seq}^{-1} & (\mathit{interleaving}) \\ AB =_\theta A \cup B & \text{if } A \times B \subseteq \mathsf{seq} \cap \mathsf{sim} & (\mathit{serialisability}) \end{array}}$$

The resulting relations $\approx_{EQ_\theta}$ and $\equiv_{EQ_\theta}$ on step sequences will respectively be denoted by $\approx_\theta$ and $\equiv_\theta$, and the set of equivalence classes of $\equiv_\theta$ which contain at least one step sequence in $\mathsf{SSEQ}_\theta = \mathbb{S}^*_\theta$, called *traces* of step sequences, by TSSEQ_θ. Moreover, the trace containing $u \in \mathsf{SSEQ}_\theta$ will be denoted by $[\![u]\!]_\theta$.

Applying the equations in EQ_θ to step sequences composed of legal steps can never produce an illegal step.

Proposition 10. *If* $\tau \in \mathsf{TSSEQ}_\theta$ *then* $\tau \subseteq \mathsf{SSEQ}_\theta$.

Proof. Follows from the fact that for an equation $AB =_\psi A \cup B$ with $A, B \in \mathbb{S}_\psi$, we have that $A \cup B \subseteq \mathsf{sim}$. $\square$

As the following result demonstrates, the entire sequentialisability is used to define the interleaving and serialisability equations.

Proposition 11. *If* $\theta = \langle \Sigma, \mathsf{sim}, \mathsf{seq} \rangle$ *is a generalised concurrency alphabet then*

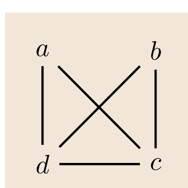
$$\mathsf{seq} = (\mathsf{seq} \cap \mathsf{seq}^{-1}) \cup (\mathsf{seq} \cap \mathsf{sim}) \subseteq \mathsf{seq}^{-1} \cup \mathsf{sim}\ .$$

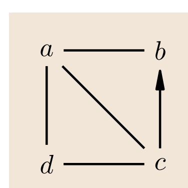
Proof. We first show that $\mathsf{seq} \subseteq \mathsf{seq}^{-1} \cup \mathsf{sim}$. Let $\langle x, y \rangle \in \mathsf{seq}$. If $\langle x, y \rangle \in \mathsf{sim}$ we are done. If $\langle x, y \rangle \notin \mathsf{sim}$ then, because $\mathsf{seq} \setminus \mathsf{sim}$ is symmetric, $\langle y, x \rangle \in \mathsf{seq}$, and so $\langle x, y \rangle \in \mathsf{seq}^{-1}$. The part $\mathsf{seq} = (\mathsf{seq} \cap \mathsf{seq}^{-1}) \cup (\mathsf{seq} \cap \mathsf{sim})$ follows immediately from $\mathsf{seq} \subseteq \mathsf{seq}^{-1} \cup \mathsf{sim}$. $\square$

The two representations of extended concurrency alphabets as fundamental concurrency alphabets and generalised concurrency alphabets, are equivalent in the sense that the traces defined are the same. We show this using the following two mappings:

$$\boxed{\begin{array}{llr} \Psi \xrightarrow{\mathsf{fca2gca}} \Theta & & \langle \Sigma, \mathsf{sim}, \mathsf{inl}, \mathsf{ser} \rangle \xmapsto{\mathsf{fca2gca}} \langle \Sigma, \mathsf{sim}, \mathsf{inl} \cup \mathsf{ser} \rangle \\ \Theta \xrightarrow{\mathsf{gca2fca}} \Psi & & \langle \Sigma, \mathsf{sim}, \mathsf{seq} \rangle \xmapsto{\mathsf{gca2fca}} \langle \Sigma, \mathsf{sim}, (\mathsf{seq} \cap \mathsf{seq}^{-1}) \setminus \mathsf{sim}, \mathsf{seq} \cap \mathsf{sim} \rangle \end{array}}$$

Example 2. The generalised concurrency alphabet corresponding to the fundamental concurrency alphabet ψ_0 of Example 1 has the following simultaneity and sequentialising relations:

$\mathsf{sim} =$ 

$\mathsf{seq} =$ 

Proposition 12. *The mappings* $\mathsf{fca2gca}$ *and* $\mathsf{gca2fca}$ *are well-defined.*

Proof. Let $\psi = \langle \Sigma, \mathsf{sim}, \mathsf{inl}, \mathsf{ser}\rangle \in \Psi$. Then $\mathsf{fca2gca}(\psi) = \langle \Sigma, \mathsf{sim}, \mathsf{inl} \cup \mathsf{ser}\rangle \in \Theta$. Indeed, $\mathsf{inl} \cup \mathsf{ser}$ is clearly irreflexive. Moreover, by $\mathsf{ser} \subseteq \mathsf{sim}$ and $\mathsf{inl} \cap \mathsf{sim} = \emptyset$, $(\mathsf{inl} \cup \mathsf{ser}) \setminus \mathsf{sim} = \mathsf{inl} \setminus \mathsf{sim} = \mathsf{inl}$. Hence $(\mathsf{inl} \cup \mathsf{ser}) \setminus \mathsf{sim}$ is symmetric as inl is.

Now, let $\theta = \langle \Sigma, \mathsf{sim}, \mathsf{seq}\rangle \in \Theta$. Then

$$\mathsf{gca2fca}(\theta) = \langle \Sigma, \mathsf{sim}, (\mathsf{seq} \cap \mathsf{seq}^{-1}) \setminus \mathsf{sim}, \mathsf{seq} \cap \mathsf{sim}\rangle \in \Psi \ .$$

Indeed, $(\mathsf{seq} \cap \mathsf{seq}^{-1}) \setminus \mathsf{sim}$ and $\mathsf{seq} \cap \mathsf{sim}$ are clearly irreflexive, $(\mathsf{seq} \cap \mathsf{seq}^{-1}) \setminus \mathsf{sim}$ is symmetric, and $\mathsf{seq} \cap \mathsf{sim} \subseteq \mathsf{sim}$. Moreover, $((\mathsf{seq} \cap \mathsf{seq}^{-1}) \setminus \mathsf{sim}) \cap \mathsf{sim} = \varnothing$. $\square$

We call mappings $\Psi \xrightarrow{f} \Theta \xrightarrow{g} \Psi$ *trace-preserving* if, for all $\psi \in \Psi$ and $\theta \in \Theta$,

$$\mathsf{TSSEQ}_{f(\psi)} = \mathsf{TSSEQ}_{\psi} \quad \text{and} \quad \mathsf{TSSEQ}_{g(\theta)} = \mathsf{TSSEQ}_{\theta} \ .$$

Theorem 1. $\Psi \xrightarrow{\mathsf{fca2gca}} \Theta \xrightarrow{\mathsf{gca2fca}} \Psi$ *are trace-preserving inverse bijections.*

Proof. To show that the mappings are inverse bijections, we show that

$$\mathsf{gca2fca} \circ \mathsf{fca2gca}(\psi) = \psi \quad \text{and} \quad \mathsf{fca2gca} \circ \mathsf{gca2fca}(\theta) = \theta \ ,$$

for all $\psi = \langle \Sigma, \mathsf{sim}, \mathsf{inl}, \mathsf{ser}\rangle \in \Psi$ and $\theta = \langle \Sigma, \mathsf{sim}, \mathsf{seq}\rangle \in \Theta$. Indeed, we have that

$$\begin{aligned}\mathsf{gca2fca} \circ \mathsf{fca2gca}(\psi) &= \mathsf{gca2fca}(\langle \Sigma, \mathsf{sim}, \mathsf{inl} \cup \mathsf{ser}\rangle)\\ &= \langle \Sigma, \mathsf{sim}, ((\mathsf{inl} \cup \mathsf{ser}) \cap (\mathsf{inl} \cup \mathsf{ser})^{-1}) \setminus \mathsf{sim}, (\mathsf{inl} \cup \mathsf{ser}) \cap \mathsf{sim}\rangle\\ &= \langle \Sigma, \mathsf{sim}, \mathsf{inl}, \mathsf{ser}\rangle \ ,\end{aligned}$$

where the last equality follows from

$$\begin{aligned}&((\mathsf{inl} \cup \mathsf{ser}) \cap (\mathsf{inl} \cup \mathsf{ser})^{-1}) \setminus \mathsf{sim}\\ &\quad = ((\mathsf{inl} \cap \mathsf{inl}^{-1}) \cup (\mathsf{ser} \cap \mathsf{inl}^{-1}) \cup (\mathsf{inl} \cap \mathsf{ser}^{-1}) \cup (\mathsf{ser} \cap \mathsf{ser}^{-1})) \setminus \mathsf{sim}\\ &\quad = (\mathsf{inl} \cap \mathsf{inl}^{-1}) \setminus \mathsf{sim} \cup (\mathsf{ser} \cap \mathsf{ser}^{-1}) \setminus \mathsf{sim} = \mathsf{inl} \cap \mathsf{inl}^{-1} = \mathsf{inl}\end{aligned}$$

and the symmetry of inl, $\mathsf{sim} \cap \mathsf{inl} = \varnothing$, and $\mathsf{ser} \subseteq \mathsf{sim}$, as well as

$$(\mathsf{inl} \cup \mathsf{ser}) \cap \mathsf{sim} = (\mathsf{inl} \cap \mathsf{sim}) \cup (\mathsf{ser} \cap \mathsf{sim}) = \mathsf{ser} \cap \mathsf{sim}$$

and $\mathsf{ser} \subseteq \mathsf{sim}$. We then observe that

$$\begin{aligned}\mathsf{fca2gca} \circ \mathsf{gca2fca}(\theta) &= \mathsf{fca2gca}(\langle \Sigma, \mathsf{sim}, (\mathsf{seq} \cap \mathsf{seq}^{-1}) \setminus \mathsf{sim}, \mathsf{seq} \cap \mathsf{sim}\rangle)\\ &= \langle \Sigma, \mathsf{sim}, ((\mathsf{seq} \cap \mathsf{seq}^{-1}) \setminus \mathsf{sim}) \cup (\mathsf{seq} \cap \mathsf{sim})\rangle\\ &= \langle \Sigma, \mathsf{sim}, \mathsf{seq}\rangle \ ,\end{aligned}$$

where the last equality follows from

$$\begin{aligned}&((\mathsf{seq} \cap \mathsf{seq}^{-1}) \setminus \mathsf{sim}) \cup (\mathsf{seq} \cap \mathsf{sim})\\ &\quad = ((\mathsf{seq} \setminus \mathsf{sim}) \cap (\mathsf{seq}^{-1} \setminus \mathsf{sim})) \cup (\mathsf{seq} \cap \mathsf{sim})\\ &\quad = (\mathsf{seq} \setminus \mathsf{sim}) \cup (\mathsf{seq} \cap \mathsf{sim}) = \mathsf{seq}\end{aligned}$$

and $\mathsf{seq} \setminus \mathsf{sim} = \mathsf{seq} \setminus \mathsf{sim} \cap \mathsf{seq}^{-1} \setminus \mathsf{sim}$ which holds because $\mathsf{seq} \setminus \mathsf{sim}$ is symmetric. To prove that fca2gca and gca2fca are trace-preserving, it suffices to show that $\mathsf{TSSEQ}_{\mathsf{fca2gca}(\psi)} = \mathsf{TSSEQ}_{\psi}$, for every $\psi = \langle \Sigma, \mathsf{sim}, \mathsf{inl}, \mathsf{ser} \rangle \in \Psi$.

Let $\mathsf{fca2gca}(\psi) = \langle \Sigma, \mathsf{sim}, \mathsf{seq} \rangle$. Then, clearly $\mathbb{S}_{\mathsf{fca2gca}(\psi)} = \mathbb{S}_{\psi}$. Moreover, $\mathsf{seq} \cap \mathsf{sim} = \mathsf{ser} \cap \mathsf{sim} = \mathsf{ser}$ as we have $\mathsf{inl} \cap \mathsf{sim} = \varnothing$ and $\mathsf{ser} \subseteq \mathsf{sim}$, and so the serialisability equations induced by the two alphabets are the same. The interleaving equations are also the same, as we have:

$$\begin{aligned} \mathsf{seq} \cap \mathsf{seq}^{-1} &= (\mathsf{inl} \cup \mathsf{ser}) \cap (\mathsf{inl} \cup \mathsf{ser})^{-1} = (\mathsf{inl} \cup \mathsf{ser}) \cap (\mathsf{inl}^{-1} \cup \mathsf{ser}^{-1}) \\ &= (\mathsf{inl} \cup \mathsf{ser}) \cap (\mathsf{inl} \cup \mathsf{ser}^{-1}) = \mathsf{inl} \cup (\mathsf{ser} \cap \mathsf{ser}^{-1}) \ . \end{aligned}$$

Hence ψ and $\mathsf{fca2gca}(\psi)$ induce the same equations over $\mathbb{S}^*_{\mathsf{fca2gca}(\psi)} = \mathbb{S}^*_{\psi}$. We can therefore conclude that $\mathsf{TSSEQ}_{\mathsf{fca2gca}(\psi)} = \mathsf{TSSEQ}_{\psi}$. □

From this point on, for ease of reference, we may refer to the traces of step sequences defined by fundamental and general concurrency alphabet as *generalised traces.*

The classification of the fundamental concurrency alphabets shown in Figure 2 can be repeated for generalised concurrency alphabets after observing that the following are the three distinct components in the Venn diagram of the relations sim and seq:

$$\boxed{\mathsf{sim} \setminus \mathsf{seq} \qquad \mathsf{sim} \cap \mathsf{seq} \qquad \mathsf{seq} \setminus \mathsf{sim}}$$

Hence, also in this case, we can distinguish eight classes of concurrency alphabets, as shown in Figure 2, where the subscripts indicate which relations are empty. Thus, for example, Θ_{ser} comprises all generalised concurrency alphabets such that $\mathsf{ser} = \varnothing$. We then obtain that the classifications for the two types of alphabets coincide.

Theorem 2. *The following are pairs of trace-preserving inverse bijections:*

$$\begin{array}{rcl} \Psi \xrightarrow{\mathsf{fca2gca}} & \Theta & \xrightarrow{\mathsf{gca2fca}} \Psi \\ \Psi_{\mathsf{sim}\setminus\mathsf{ser}} \xrightarrow{\mathsf{fca2gca}} & \Theta_{\mathsf{sim}\setminus\mathsf{seq}} & \xrightarrow{\mathsf{gca2fca}} \Psi_{\mathsf{sim}\setminus\mathsf{ser}} \\ \Psi_{\mathsf{inl}} \xrightarrow{\mathsf{fca2gca}} & \Theta_{\mathsf{seq}\setminus\mathsf{sim}} & \xrightarrow{\mathsf{gca2fca}} \Psi_{\mathsf{inl}} \\ \Psi_{\mathsf{ser}} \xrightarrow{\mathsf{fca2gca}} & \Theta_{\mathsf{sim}\cap\mathsf{seq}} & \xrightarrow{\mathsf{gca2fca}} \Psi_{\mathsf{ser}} \\ \Psi_{\mathsf{inl}\cup\mathsf{ser}} \xrightarrow{\mathsf{fca2gca}} & \Theta_{\mathsf{seq}} & \xrightarrow{\mathsf{gca2fca}} \Psi_{\mathsf{inl}\cup\mathsf{ser}} \\ \Psi_{\mathsf{sim}} \xrightarrow{\mathsf{fca2gca}} & \Theta_{\mathsf{sim}} & \xrightarrow{\mathsf{gca2fca}} \Psi_{\mathsf{sim}} \\ \Psi_{(\mathsf{sim}\setminus\mathsf{ser})\cup\mathsf{inl}} \xrightarrow{\mathsf{fca2gca}} & \Theta_{(\mathsf{sim}\setminus\mathsf{seq})\cup(\mathsf{seq}\setminus\mathsf{sim})} & \xrightarrow{\mathsf{gca2fca}} \Psi_{(\mathsf{sim}\setminus\mathsf{ser})\cup\mathsf{inl}} \\ \Psi_{\mathsf{sim}\cup\mathsf{inl}} \xrightarrow{\mathsf{fca2gca}} & \Theta_{\mathsf{sim}\cup\mathsf{seq}} & \xrightarrow{\mathsf{gca2fca}} \Psi_{\mathsf{sim}\cup\mathsf{inl}} \end{array}$$

Proof. The result follows from Theorem 1, the definitions of the mappings fca2gca and gca2fca, $\mathsf{sim} \setminus \mathsf{ser} = \mathsf{sim} \setminus \mathsf{seq}$, $\mathsf{inl} = \mathsf{sim} \cap \mathsf{seq}$, and $\mathsf{ser} = \mathsf{seq} \setminus \mathsf{sim}$. □

Thus we can conclude that the two families of fundamental concurrency alphabets and generalised concurrency alphabets are essentially the same. However, since the latter type involves only two rather than three relations it is often in technical considerations more convenient to deal with.

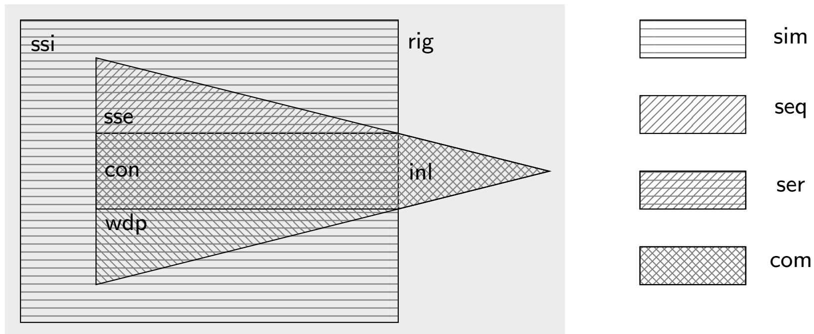


Fig. 3. Relationships involving actions in generalised concurrency alphabets.

4.4 Another look at concurrency alphabets

In the last part of this section, we take another look at the structure of concurrency alphabets. Let us consider a fundamental concurrency alphabet $\theta = \langle \Sigma, \mathsf{sim}, \mathsf{inl}, \mathsf{ser} \rangle \in \Theta$, and a generalised concurrency alphabet $\psi = \langle \Sigma, \mathsf{sim}, \mathsf{seq} \rangle \in \Psi$. Moreover, assume that they correspond to each other, i.e., $\theta = \mathsf{fca2gca}(\psi)$. We then single out six semantically meaningful relationships between pairs of actions which form a partition of $\Sigma \times \Sigma$ (see Figure 3, and [18] for a successful use of similar partition in the case of comtraces):

$\boxed{\mathsf{ssi} = \mathsf{sim} \setminus (\mathsf{seq} \cup \mathsf{seq}^{-1}) = \mathsf{sim} \setminus (\mathsf{ser} \cup \mathsf{ser}^{-1})}$

is *strong simultaneity* allowing a pair of actions to be executed simultaneously, and disallowing serialisation and interleaving.

$\boxed{\mathsf{sse} = (\mathsf{seq} \setminus \mathsf{seq}^{-1}) \cap \mathsf{sim} = \mathsf{ser} \setminus \mathsf{ser}^{-1}}$

is *semi-serialisability* allowing a pair of simultaneously executed actions to be executed in the order given, but not in the reverse order.

$\boxed{\mathsf{con} = \mathsf{seq} \cap \mathsf{seq}^{-1} \cap \mathsf{sim} = \mathsf{ser} \cap \mathsf{ser}^{-1}}$

is *concurrency* identifying actions which can be executed simultaneously as well as in any order.

$\boxed{\mathsf{wdp} = (\mathsf{seq}^{-1} \setminus \mathsf{seq}) \cap \mathsf{sim} = \mathsf{ser}^{-1} \setminus \mathsf{ser}}$

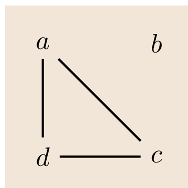
is *weak dependence* which is an inverse of semi-serialisability.

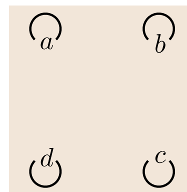
$\boxed{\mathsf{rig} = (\Sigma \times \Sigma) \setminus (\mathsf{sim} \cup (\mathsf{seq} \cap \mathsf{seq}^{-1})) = (\Sigma \times \Sigma) \setminus (\mathsf{sim} \cup \mathsf{inl})}$

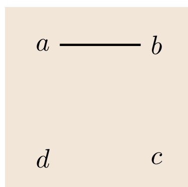
is *rigid order* allowing neither simultaneity nor changing of the order of actions.

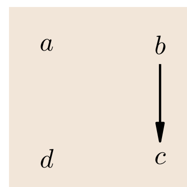
$\boxed{\mathsf{inl} = (\mathsf{seq} \cap \mathsf{seq}^{-1}) \setminus \mathsf{sim}}$
is *interleaving* as before.

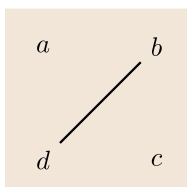
Example 3. For the generalised concurrency alphabet of Example 1, the relations derived above are as follows:

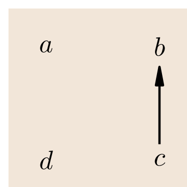
$\mathsf{con} =$ 

$\mathsf{rig} =$ 

$\mathsf{inl} =$ 

$\mathsf{wdp} =$ 

$\mathsf{ssi} =$ 

$\mathsf{sse} =$ 

Hence a and b are the only truly interleaved actions, while b and d are the only actions whose serialisation and interleaving is disallowed (this does not prevent b and d from occurring in the same step). The rigid order, which plays the role of dependence in the Mazurkiewicz trace theory, is implied by label-linearity and does not involve any pair of different actions. $\diamond$

5 Extending causal structures

This section describes the labelled relational structures, called *order structures*, which we will use to represent the observational and causal relationships in the observations of behaviours of concurrent systems. Also introduced are *layered order structures* that represent individual step sequence observations. The essential goal is to provide relational structures matching generalised traces and step sequences in the same way as partial orders match Mazurkiewicz traces whereas total orders match sequences of action occurrences.

5.1 Order structures

To model the relationships between occurrences of actions, we require relational structures with two specific properties.

Definition 3 (order structure). *A relational structure os is an* order structure *if it is both separable and label-ordered. The family of all order structures will be denoted by* OS*. Moreover, the relations* $\rightleftharpoons_{os}$ *and* $\sqsubset_{os}$ *will be respectively called* mutex *and* weak causality*.* $\diamond$

Intuitively, the domain Δ of an order structure $rs = \langle \Delta, \rightleftharpoons, \sqsubset, \ell \rangle$, is the set of events that have happened, $x \rightleftharpoons y$ means that x occurred *not simultaneously* with y, and $x \sqsubset y$ that x occurred *not later* than y, i.e., *before or simultaneously* with y. Hence if both $x \sqsubset y$ and $x \rightleftharpoons y$ hold, then x must have occurred *before* y. We will therefore refer to the intersection of $\sqsubset$ and $\rightleftharpoons$ as *causality* (or *precedence*), denoting it by $\prec$. Note that $x \sqsubset y \sqsubset x$ intuitively means that x and y were observed as *simultaneous*. The requirement of separability then excludes situations where events forming a weak causality cycle (captured by $\sqsubset^{\circledast}$) are also involved in the mutex relationship. (Referring to the set-up of Mazurkiewicz traces, order structures correspond to acyclic relations.) The labelling function ℓ associates an action with each event, with distinct events corresponding to distinct occurrences (or executions) of actions. Label-orderedness together with separability guarantees that all events labelled by the same action are totally ordered, i.e., the structure is *label-linear* (see Proposition 3).

Note that a direct predecessor of order structures were the *stratified order structures* (where $\rightleftharpoons$ is included in $\sqsubset$), introduced independently in [2] and [5], and then applied, e.g., in [10, 11].

5.2 Layered order structures

An order structure representing a single observation (a step sequence) has to have all the observational relationships between events determined, i.e., it needs to be $\lhd$-maximal within the set of order structures.

Definition 4 (saturated order structure). *An order structure os is* saturated *if* $\mathsf{ext}(os) \cap \mathsf{OS} = \{os\}$*. The family of all saturated order structures will be denoted by* SOS*.* $\diamond$

In the model of Mazurkiewicz traces, single observations are represented by total orders which can be thought of as saturated partial orders (as adding any additional ordering between elements destroys acyclicity). In the case of a saturated order structure, adding extra mutex or weak causality relations between events destroys separability. Note that, in the original definition of saturated order structures in [4], label-orderedness was not an issue as only unlabelled structures were considered there.

Knowing only that an order structure is saturated is not very useful when it comes to proofs and understanding of other properties. Therefore, we will now provide an axiomatic description of saturated order structures in terms of the characteristic properties of their component relations.

Definition 5 (layered order structure). *A* layered order structure *(LO-structure) is a relational structure*

$$\boxed{los = \langle \Delta, \rightleftharpoons, \sqsubset, \ell \rangle} \quad \textit{satisfying}$$

$$\begin{array}{rrcll}
x \neq y \ \wedge & x \sqsubset z \sqsubset y & \Longrightarrow & x \sqsubset y & : \ L1 \\
 & x \rightleftharpoons y & \Longrightarrow & x \sqsubset^{sym} y & : \ L2 \\
x \neq y \ \wedge & x \not\rightleftharpoons y & \Longleftrightarrow & x \sqsubset y \sqsubset x & : \ L3 \\
x \neq y \ \wedge & \ell(x) = \ell(y) & \Longrightarrow & x \rightleftharpoons y & : \ L4
\end{array}$$

The family of all layered order structures will be denoted by LOS. ⋄

Intuitively, e.g., *L2* means that if x and y are not simultaneous, then one of them must be before the other. Moreover, together *L2* and *L4* imply label-orderedness.

Proposition 13. $\mathsf{LOS} = \mathsf{SOS}$.

Proof. Let $los = \langle \Delta, \rightleftharpoons, \sqsubset, \ell \rangle \in \mathsf{LOS}$. First we show that los is separable:

- Suppose that $x \sqsubset x$. Then $x \sqsubset x \sqsubset x$ and so, by *L3*, $x \neq x$ which produces a contradiction. Hence $\sqsubset$ is irreflexive. Therefore, by *L2*, $\rightleftharpoons$ is also irreflexive.
- Suppose that $x \not\rightleftharpoons y$ and $x \neq y$. Then, by *L3*, we have $x \sqsubset y \sqsubset x$ and thus also $y \sqsubset x \sqsubset y$ which in turn implies $y \not\rightleftharpoons x$. Hence $\not\rightleftharpoons$ is symmetric.
- Suppose that $x \sqsubset^{\circledast} y$. If $x = y$ then, by the irreflexivity of $\rightleftharpoons$, we have $x \not\rightleftharpoons y$. If $x \neq y$ then, by repeated application of *L1*, $x \sqsubset y \sqsubset x$. Hence, by *L3*, $x \not\rightleftharpoons y$ and so we can conclude that $\rightleftharpoons \cap \sqsubset^{\circledast} = \varnothing$.

As a result, los is separable. Moreover, los is label-ordered. Indeed, suppose that $x \neq y$ and $\ell(x) = \ell(y)$. Then, by *L4*, $x \rightleftharpoons y$ and so, by *L2*, we have $x \sqsubset^{sym} y$. Thus $x \prec^{sym} y$.

We can therefore conclude that $los \in \mathsf{OS}$. To show that $los \in \mathsf{SOS}$, suppose that $os \neq los$ is an order structure such that $los \lhd os$. Then there must exist $x, y \in \Delta$ such that one of the following holds:

- $x \rightleftharpoons_{os} y$ and $x \not\rightleftharpoons y$. Since $\rightleftharpoons_{os}$ is irreflexive, $x \neq y$. Hence, by *L3*, $x \sqsubset y \sqsubset x$. Therefore, by $los \lhd os$, $x \sqsubset^{\circledast}_{os} y$ which, together with $x \rightleftharpoons_{os} y$, contradicts the separability of os.
- $x \sqsubset_{os} y$ and $x \not\sqsubset y$. Since $\sqsubset_{os}$ is irreflexive, $x \neq y$. Hence, by *L3*, we have $x \rightleftharpoons y$. Thus, by *L2* and $x \not\sqsubset y$, we obtain $y \sqsubset x$. Therefore, by $los \lhd os$, $x \sqsubset^{\circledast}_{os} y$ and $x \rightleftharpoons_{os} y$, contradicting the separability of os.

Since in both cases we obtained a contradiction, los is a saturated order structure.

Conversely, let $os = \langle \Delta, \rightleftharpoons, \sqsubset, \ell \rangle \in \mathsf{SOS}$. We first show that if $x \neq y$ then:

(a) $x \not\rightleftharpoons y$ implies $x \sqsubset^{+} y \sqsubset^{+} x$.
(b) $x \not\sqsubset y$ implies $y \sqsubset^{+} x$ and $x \rightleftharpoons y$.

(a) We first observe that $y \not\rightleftharpoons x$, as $\rightleftharpoons$ is symmetric. We then consider a relational structure os' obtained from os by adding the pair $\langle x, y\rangle$ to $\rightleftharpoons$. Since $os' \neq os$ and $os \lhd os'$, it follows from $os \in \mathsf{SOS}$ that $os' \notin \mathsf{OS}$. We then observe that in such a case $\langle x, y\rangle$ must belong to $\rightleftharpoons_{os'} \cap \sqsubset^{\circledast}_{os'}$. Hence, by $\sqsubset^{\circledast}_{os'} = \sqsubset^{\circledast}$, we obtain that $x \sqsubset^{+} y \sqsubset^{+} x$.

(b) We consider a relational structure os' obtained from os by adding the pair $\langle x, y\rangle$ to $\sqsubset$. As in the case of *(a)*, $os' \notin \mathsf{OS}$. We then observe that in such a case there is a pair $\langle w, u\rangle$ belonging to $\rightleftharpoons_{os'} \cap \sqsubset^{\circledast}_{os'}$. Clearly, $w \rightleftharpoons u$ and the only way that $w \sqsubset^{\circledast}_{os'} u$ holds is that we created a cycle through adding $\langle x, y\rangle$ to $\sqsubset$. Hence we must have had $y \sqsubset^{+} x$. Suppose that $x \not\rightleftharpoons y$. Then, by *(a)*, $x \sqsubset^{\circledast} y$ and so $w \sqsubset^{\circledast} u$ which produces a contradiction with the separability of os. Hence $x \rightleftharpoons y$, and so (b) holds.

We will now show that os is an LO-structure, by checking the satisfaction of the defining conditions *L1*–*L4*:

- Suppose that $x \neq y$ and $x \sqsubset z \sqsubset y$ and $x \not\sqsubset y$. Then, by (b), $y \sqsubset^{+} x$ and $x \rightleftharpoons y$. Thus $y \sqsubset^{+} x \sqsubset z \sqsubset y$, and so $\langle x, y\rangle$ belongs to $\rightleftharpoons \cap \sqsubset^{\circledast}$, contradicting the separability of os. As a result, os satisfies *L1*.
- Suppose that $x \rightleftharpoons y$ and $x \not\sqsubset^{sym} y$ (i.e., $x \not\sqsubset y$ and $y \not\sqsubset x$). Since $\rightleftharpoons$ is irreflexive, $x \neq y$. Then, by (a), $y \sqsubset^{+} x$ and $x \sqsubset^{+} y$. Thus $y \sqsubset^{+} x \sqsubset^{+} y$, and so $\langle x, y\rangle$ belongs to $\rightleftharpoons \cap \sqsubset^{\circledast}$, contradicting the separability of os. As a result, os satisfies *L2*.
- Suppose first that $x \neq y$ and $x \not\rightleftharpoons y$. Then, by (a), $x \sqsubset^{+} y \sqsubset^{+} x$, and so, by an already demonstrated *L1*, $x \sqsubset y \sqsubset x$. Conversely, suppose that $x \sqsubset y \sqsubset x$. Then, by the irreflexivity of $\sqsubset$, we have $x \neq y$, and, by $\rightleftharpoons \cap \sqsubset^{\circledast} = \varnothing$, we have $x \not\rightleftharpoons y$. As a result, os satisfies *L3*.
- Suppose that $x \neq y$, $\ell(x) = \ell(y)$, and $x \not\rightleftharpoons y$. Then $x \not\prec^{sym} y$, contradicting the label-orderedness of os. As a result, os satisfies *L4*.

Hence we can conclude that $os \in \mathsf{LOS}$. $\square$

5.3 Invariants and histories

Within the order-theoretic part of Mazurkiewicz' approach, there are two ways in which one can represent concurrent behaviour: (i) by means of a causal partial order *po* (or a causal *invariant* in the terminology of [6]); and (ii) through a set of total orders T which are the sequential observations of *po* (or a *history* in the terminology of [6]). These two representations are in one-to-one correspondence; more precisely, T is obtained by linearising *po*, and *po* can be obtained from T by intersecting the total orders it contains. This combination of invariant/history has been taken up in [6], where a general notion of history and underlying invariants have been proposed. We will revisit this general set-up for the model of order structures.

Following the general approach, we consider two ways of representing a history. An order structure (a dependence graph), typically non-saturated, captures the causal invariants underlying the history, whereas a set of saturated order structures (i.e., LO-structures), captures the observations of the history. Of course, not any combination of LO-structures represents a concurrent history. Below we assume that all LO-structures involved have at least *the same action occurrences* and *the same ordering* of the occurrences of any given action.

Definition 6 (LO-structure set). *An* LO-structure set *(*LOS*-set) is a label-ordered consistent set of* LO*-structures. The family of* LOS*-sets will be denoted by* LOSS*.* ◇

In other words, the LO-structures belonging to an LOS-set share their domain and, in addition, induce the same total ordering on events labelled by any given action (see Propositions 3, 5 and 6).

To move between LOS-sets (histories) and order structures (invariants) we use the operations of *intersection*, $\mathsf{loss2os}$, and *saturation*, $\mathsf{os2loss}$:

$$\boxed{\begin{array}{llll} \mathsf{LOSS} \xrightarrow{\mathsf{loss2os}} \mathsf{OS} & \qquad & loss \xmapsto{\mathsf{loss2os}} \bigcap loss \\ \mathsf{OS} \xrightarrow{\mathsf{os2loss}} \mathsf{LOSS} & \qquad & os \xmapsto{\mathsf{os2loss}} \mathsf{ext}(os) \cap \mathsf{LOS} \end{array}}$$

Proposition 14. *The mappings* $\mathsf{loss2os}$ *and* $\mathsf{os2loss}$ *are well-defined.*

Proof. Suppose that $loss \in \mathsf{LOSS}$ and $os = \mathsf{loss2os}(loss) = \bigcap loss$. Then os is separable by Proposition 5(2), and its label-orderedness follows from the definitions.

Suppose now that $os \in \mathsf{OS}$ and $loss = \mathsf{os2loss}(os) = \mathsf{ext}(os) \cap \mathsf{LOS}$. From Prop.7 and Th.3 in [4], it follows that $loss \neq \varnothing$. Clearly, $loss$ is label-ordered and consistent due to the definition of $\mathsf{ext}(os)$. □

We are now in a position to state what it means that an order structure is an invariant, and that an LOS-set is a history.

Definition 7 (invariants & histories). *An order structure ios is an* invariant order structure *(*IO*-structure), and an* LOS*-set hloss is a* history LOS-set *(*HLOS*-set) if, respectively, the following hold:*[5]

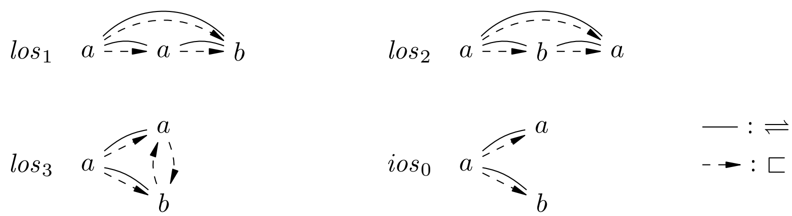
$$\boxed{ios = \mathsf{loss2os} \circ \mathsf{os2loss}(ios) \quad \textit{and} \quad hloss = \mathsf{os2loss} \circ \mathsf{loss2os}(hloss)}$$

The families of all invariant order structures and all histories will be denoted by IOS *and* HLOSS*, respectively.* ◇

Note that $ios = \mathsf{loss2os} \circ \mathsf{os2loss}(ios)$ is a version of Szpilrajn's property [20], which states that a poset is the intersection of its total order extensions, and plays a key role in the model of Mazurkiewicz traces.

Example 4. Consider three LO-structures, los_i ($i = 1, 2, 3$), and an order structure, ios_0, depicted below:

[5] Note that $\mathsf{os2loss}(ios) \neq \varnothing$ holds by Proposition 14.



Then $hloss_0 = \{los_1, los_2, los_3\}$ is a history LOS-set and ios_0 is an invariant order structure such that $ios_0 = \mathsf{loss2os}(hloss_0)$ and $hloss_0 = \mathsf{os2loss}(ios_0)$. $\diamond$

IO-structures are the causal invariants in the realm of order structures and, according to the next result, their sets of saturated extensions are concurrent histories.

Theorem 3. $\mathsf{IOS} \xrightarrow{\mathsf{os2loss}} \mathsf{HLOSS} \xrightarrow{\mathsf{loss2os}} \mathsf{IOS}$ *are inverse bijections.*

Proof. Suppose that $ios \in \mathsf{IOS}$ and $loss = \mathsf{os2loss}(ios)$. Then, by Definition 7, $ios = \mathsf{loss2os} \circ \mathsf{os2loss}(ios)$. Hence $loss = \mathsf{os2loss}(ios) = \mathsf{os2loss} \circ \mathsf{loss2os} \circ \mathsf{os2loss}(ios) = \mathsf{os2loss} \circ \mathsf{loss2os}(loss)$ and so, by Definition 7, $loss \in \mathsf{HLOSS}$.

Suppose now that $hloss \in \mathsf{HLOSS}$ and $os = \mathsf{loss2os}(hloss)$. Then, by Definition 7, $\mathsf{os2loss}(os) = hloss$. Hence $os = \mathsf{loss2os} \circ \mathsf{os2loss}(os)$ and $os \in \mathsf{IOS}$. $\square$

5.4 GMO-structures

An axiomatic characterisation of invariant order structures without domain labellings was introduced in [4]. In the next definition we recall this characterisation and add (Axiom *G7*) to ensure label-linearity.

Definition 8 (generalised mutex order structure). *A* generalised mutex order structure *(*GMO*-structure) is a relational structure*

$$\boxed{gmos = \langle \Delta, \rightleftharpoons, \sqsubset, \ell \rangle} \quad \textit{satisfying}$$

$$\begin{array}{rcll}
 & & x \not\sqsubset x & : \; G1 \\
x \neq y \;\wedge\; \; x \sqsubset z \sqsubset y & \Longrightarrow & x \sqsubset y & : \; G2 \\
x \rightleftharpoons y & \Longrightarrow & y \rightleftharpoons x \neq y & : \; G3 \\
x \prec z \sqsubset y \;\vee\; \; x \sqsubset z \prec y & \Longrightarrow & x \rightleftharpoons y & : \; G4 \\
z \rightleftharpoons y \;\wedge\; \; z \sqsubset x \sqsubset z & \Longrightarrow & x \rightleftharpoons y & : \; G5 \\
z \rightleftharpoons z' \;\wedge\; x \sqsubset z \sqsubset y \;\wedge\; \; x \sqsubset z' \sqsubset y & \Longrightarrow & x \rightleftharpoons y & : \; G6 \\
x \neq y \;\wedge\; \ell(x) = \ell(y) & \Longrightarrow & x \prec^{sym} y & : \; G7
\end{array}$$

The family of all generalised mutex order structures will be denoted by GMOS. $\diamond$

Order structures are like dependence graphs (acyclic relations) in the model of Mazurkiewicz traces, which need to be transitively closed in order to provide full information, e.g., about event precedence, in the form of partial orders. We therefore need a suitable notion of closure for order structures. Again, such a notion for order structures without domain labellings was introduced in [4], as recalled below (note that domain labelling does not play any role in this purely order-theoretic definition).

An *order structure closure* (OS-closure) is a mapping given by:

$$\boxed{\begin{array}{c} \mathsf{OS} \xrightarrow{\mathsf{os2gmos}} \mathsf{GMOS} \\ \langle \Delta, \rightleftharpoons, \sqsubset, \ell \rangle \xmapsto{\mathsf{os2gmos}} \langle \Delta, \sqsubset^{\circledast} \circ \big(\rightleftharpoons \cup (\sqsubset^{*} \circ_{\rightleftharpoons} \sqsubset^{*})^{sym} \big) \circ \sqsubset^{\circledast}, \sqsubset^{\curlywedge}, \ell \rangle \end{array}}$$

where $\circ_{\rightleftharpoons}$ and $\sqsubset^{\curlywedge}$ are defined as in Section 3 (see also Remark 1). Intuitively, the derived weak causality, $\sqsubset^{\curlywedge}$, captures the fact that weak causality is transitive. The first component of the new mutex, $\sqsubset^{\circledast} \circ \rightleftharpoons \circ \sqsubset^{\circledast}$, captures the fact that if we have two clusters of simultaneous events, and there is a pair of events in these two clusters which is non-simultaneous, then the same is true of all the pairs of events coming from these clusters (see also Axiom *G5* in Definition 8). The other component, $\sqsubset^{*} \circ_{\rightleftharpoons} \sqsubset^{*}$, captures the cross-like propagation of the mutex relationship (see also Axiom *G6*).

It is easy to check that GMO-structures are a class of order structures, and the paper [4] developed some of their key properties.

Theorem 4 (cf. Prop.7, Prop.8 and Thm.3 in [4]).

1. *If* $gmos \in \mathsf{GMOS}$*, then* $\mathsf{os2loss}(gmos) \neq \varnothing$ *and* $gmos = \mathsf{loss2os} \circ \mathsf{os2loss}(gmos)$.
2. $\mathsf{os2gmos}$ *is a structure-closure operator from* OS *to* GMOS. $\diamond$

We can now show that GMO-structures provide an alternative, axiomatic characterisation of invariant order structures.

Theorem 5. $\mathsf{GMOS} = \mathsf{IOS}$.

Proof. $\mathsf{GMOS} \subseteq \mathsf{IOS}$ holds by Definition 7 and Theorem 4(1).

Suppose that $ios \in \mathsf{IOS}$. Let $gmos = \mathsf{os2gmos}(os) \in \mathsf{GMOS}$. By Proposition 2 and Theorem 4(2), $\mathsf{os2loss}(ios) = \mathsf{os2loss}(gmos)$. Hence $\mathsf{loss2os} \circ \mathsf{os2loss}(ios) = \mathsf{loss2os} \circ \mathsf{os2loss}(gmos)$ and so, by Definition 7 and Theorem 4(1), $ios = gmos \in \mathsf{GMOS}$. Thus $\mathsf{IOS} \subseteq \mathsf{GMOS}$. $\square$

We also obtain that OS-closure is the only way in which order structures can be closed to yield invariant order structures.

Theorem 6. OS*-closure is the unique structure-closure operator from* OS *to* IOS.

Proof. By Theorems 4(2) and 5, OS-closure is a structure-closure operator from OS to IOS.

Suppose $\mathsf{OS} \xrightarrow{\mathsf{cls}} \mathsf{IOS}$ is a structure-closure operator. Let $os \in \mathsf{OS}$. Then $\mathsf{os2loss}(\mathsf{cls}(os)) = \mathsf{os2loss}(os) = \mathsf{os2loss}(\mathsf{os2gmos}(os))$, by Proposition 2($ii$) and Theorem 4($ii$). Hence, by $\mathsf{cls}(os) \in \mathsf{IOS}$ and $\mathsf{os2gmos}(os) \in \mathsf{IOS}$, we obtain $\mathsf{cls}(os) = \mathsf{os2gmos}(os)$. $\square$

In this way, we have ended our search for general relational structures corresponding to causal partial orders, and the general notion of invariant order structure and concurrent history.

Theorem 7. *The diagram in Figure 4 commutes.*

Proof. Follows from Theorem 3 and 6. □

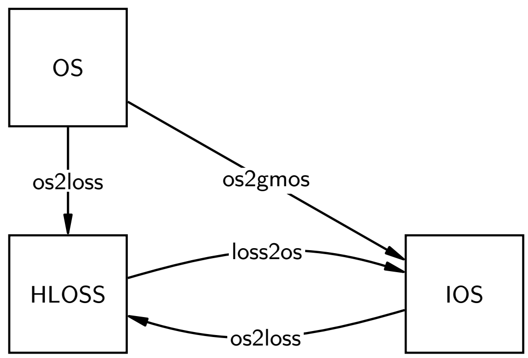


Fig. 4. Behaviour diagram for order structures.

6 Generalised traces and extended causal structures

We finally join together the two lines of our discussion, one concerned with generalisations of traces, and the other dealing with extensions of causal partial orders.

Let $\theta = \langle \Sigma, \mathsf{sim}, \mathsf{seq} \rangle$ be a generalised concurrency alphabet **fixed** throughout this section.

6.1 Step sequences and order structures

In this subsection, we first formally establish the correspondence between step sequences from SSEQ_θ and their underlying saturated order structures (i.e., layered order structures by Proposition 13). This corresponds to the way sequences can be interpreted as total orders. It, moreover, makes it possible to lift the notion of a trace to the level of LO-structures which will allow us later to discuss the equivalence of step sequences in terms of order structures.

By Proposition 4 isomorphisms between label-linear relational structures are unique and so we are free to choose the names of the elements that will carry the action names as labels. We focus on order structures whose domains can be seen as a set of events which occurred during an execution of a concurrent system. A set $\Delta \subseteq \Sigma \times \mathbb{N}$ is an *event domain* if there is a mapping $\epsilon : \Sigma \to \mathbb{N}$ such that

$$\Delta = \{\langle a, i\rangle \mid a \in \Sigma \wedge 1 \leq i \leq \epsilon(a)\} .$$

We will further assume that $\langle a, i\rangle \overset{\ell}{\mapsto} a$ is the default labelling for event domains. Note that $occ(u)$ is an event domain, for every step sequence $u \in \mathsf{SSEQ}_\theta$.

Given the generalised concurrency alphabet θ as above, an LO-structure $los = \langle \Delta, \rightleftharpoons, \sqsubset, \ell\rangle$ is *consistent* (with θ) if Δ is an event domain and, for all distinct $\langle a, i\rangle, \langle a, j\rangle, \langle b, k\rangle \in \Delta$:

$$\begin{array}{rcl} \langle a, i\rangle \prec \langle a, j\rangle & \iff & i < j \\ \langle a, i\rangle \sqsubset^{\circledast} \langle b, k\rangle & \implies & \langle a, b\rangle \in \mathsf{sim} . \end{array} \tag{3}$$

We denote this by $los \in \mathsf{LOS}_\theta$. In other words, in a consistent layered order structure, consecutive occurrences of events with the same label are totally ordered, and the labels of events that occur simultaneously, denote actions that are simultaneous according to θ.

Consistent LO-structures correspond exactly to the step sequences in SSEQ_θ as we proceed now to prove. First we show how such structures can be interpreted as sequences of sets of simultaneous events.

Proposition 15. *Let $los = \langle \Delta, \rightleftharpoons, \sqsubset, \ell\rangle \in \mathsf{LOS}_\theta$. Then there is a unique sequence $\tau_{los} = \Delta_1 \ldots \Delta_k$ such that:*

1. *$\Delta_1, \ldots, \Delta_k$ is a partition of Δ satisfying*
$$\rightleftharpoons\, = \bigcup_{i \neq j} \Delta_i \times \Delta_j \qquad \sqsubset\, = \bigcup_{i \leq j} \Delta_i \times \Delta_j \setminus id_\Delta \qquad \prec\, = \bigcup_{i < j} \Delta_i \times \Delta_j .$$
2. *$\Delta_1, \ldots, \Delta_k$ are the equivalence classes of $\sqsubset^{\circledast}$.*

Proof. Let $\mathcal{X}$ be the set of equivalence classes of $\sqsubset^{\circledast}$. For distinct $X, Y \in \mathcal{X}$, we define $X \dot{\rightleftharpoons} Y$ and $X \dot{\sqsubset} Y$ if, respectively, $(X \times Y) \cap \rightleftharpoons \neq \varnothing$ and $(X \times Y) \cap \sqsubset \neq \varnothing$. We then show that, for distinct $X, Y \in \mathcal{X}$, we have the following:

$$\begin{array}{llcl} (i) & X \dot{\sqsubset} Y & \implies & X \times Y \subseteq \sqsubset \\ (ii) & X \neq Y & \implies & X \times Y \subseteq \rightleftharpoons \\ (iii) & X \dot{\sqsubset} Y & \implies & \neg Y \dot{\sqsubset} X \\ (iv) & X \neq Y & \implies & X \dot{\sqsubset}^{sym} Y . \end{array}$$

Let $\alpha \in X$ and $\beta \in Y$. Since $X \neq Y$, also $\alpha \neq \beta$.

(i) If $X \dot{\sqsubset} Y$, then there exist $\gamma \in X$ and $\delta \in Y$ such that $\gamma \sqsubset \delta$ which together with $\alpha \neq \beta$ implies by *L1* that $\alpha \sqsubset \beta$.

(ii) If $\alpha \not\rightleftharpoons \beta$ then $\alpha \neq \beta$ implies, by *L3*, that $\alpha \sqsubset \beta \sqsubset \alpha$, a contradiction.

(iii) Follows from the maximality of $\sqsubset^{\circledast}$.

(iv) We have $\alpha \neq \beta$. If $\alpha \not\rightleftharpoons \beta$ then, by *L3*, $\alpha \sqsubset \beta \sqsubset \alpha$. Hence $X \dot{\sqsubset} Y \dot{\sqsubset} X$ which contradicts (iii). Thus we have $\alpha \rightleftharpoons \beta$ and so $\alpha \sqsubset^{sym} \beta$, by *L2*.

Now define $\dot{\prec} = \dot{\sqsubset} \cap \dot{\rightleftharpoons}$. From what we have just shown it follows that $\dot{\prec}$ is a total order relation over $\mathcal{X}$. Moreover, the order in which the equivalence classes of $\sqsubset^{\circledast}$ are ordered by $\dot{\prec}$ gives the desired sequence and verifies its uniqueness. $\square$

The unique sequence $\tau_{los} = \Delta_1 \ldots \Delta_k$ shown to exist in Proposition 15 will be called the *layer decomposition* of *los*. This decomposition defines through its labeling a step sequence associated with τ.

Proposition 16. *Let $\tau_{los} = \Delta_1 \ldots \Delta_k$ be the layer decomposition of $los \in \mathsf{LOS}_\theta$. Then, for all $1 \leq i \leq k$, labelling ℓ is injective on Δ_i and $\ell(\Delta_i) \in \mathbb{S}_\theta$.*

Proof. Let $i \leq k$ and suppose that $\alpha, \beta \in \Delta_i$, $\alpha \neq \beta$ and $\ell(\alpha) = \ell(\beta)$. Then, by *L4*, $\alpha \rightleftharpoons \beta$. Hence $\alpha \not\sqsubset^{\circledast} \beta$ as, by Proposition 13, *los* is an order structure, and so it is separable. We therefore obtained a contradiction with Proposition 15(2).

The second part follows from Proposition 15(2) and Eq.(3). $\square$

Thus a consistent LO-structure corresponds to a sequence of layers, each layer comprising events which can be seen as a valid step according to the alphabet θ. We now take advantage of this observation to establish the full correspondence between LO-structures in LOS_θ and step sequences in SSEQ_θ, using two mappings:

$$\boxed{\begin{array}{rlcrl} \mathsf{LOS}_\theta & \xrightarrow{\mathsf{los2sseq}} \mathsf{SSEQ}_\theta & \qquad & \mathsf{SSEQ}_\theta & \xrightarrow{\mathsf{sseq2los}} \mathsf{LOS}_\theta \\ los & \xmapsto{\mathsf{los2sseq}} \ell(\tau_{los}) & & u & \xmapsto{\mathsf{sseq2los}} \langle \Delta, \rightleftharpoons, \sqsubset, \ell \rangle \end{array}}$$

where $\Delta = occ(u)$ and, for all $\alpha, \beta \in occ(u)$ with $pos_u(\alpha) = k$ and $pos_u(\beta) = m$:

$$\begin{array}{lll} \alpha \rightleftharpoons \beta & \textit{if} & k \neq m \\ \alpha \sqsubset \beta & \textit{if} & k \leq m \ \wedge \ \alpha \neq \beta\ . \end{array} \tag{4}$$

Proposition 17. *The mappings $\mathsf{LOS}_\theta \xrightarrow{\mathsf{los2sseq}} \mathsf{SSEQ}_\theta$ and $\mathsf{SSEQ}_\theta \xrightarrow{\mathsf{sseq2los}} \mathsf{LOS}_\theta$ are well-defined.*

Proof. The first part follows from Proposition 16. To show the second part, we proceed as follows.

Suppose that $u \in \mathsf{SSEQ}_\theta$ and $los = \mathsf{sseq2los}(u) = \langle \Delta, \rightleftharpoons, \sqsubset, \ell \rangle$. First we demonstrate that $los \in \mathsf{LOS}$ by showing that the axioms (*L1*)–(*L4*) hold.

L1 : Suppose that $\alpha \neq \beta$ and $\alpha \sqsubset \gamma \sqsubset \beta$. By Eq.(4), we have $pos_u(\alpha) \leq pos_u(\gamma) \leq pos_u(\beta)$. Hence $pos_u(\alpha) \leq pos_u(\beta)$ and so, by Eq.(4), $\alpha \sqsubset \beta$.

L2 : Suppose that $\alpha \rightleftharpoons \beta$. By Eq.(4), we have $pos_u(\alpha) \neq pos_u(\beta)$ and so also $\alpha \neq \beta$. Hence, by Eq.(4), $\alpha \sqsubset^{sym} \beta$.

L3 : Suppose that $\alpha \neq \beta$ and $\alpha \not\rightleftharpoons \beta$. Then, by Eq.(4), $pos_u(\alpha) = pos_u(\beta)$. Hence, by Eq.(4), $\alpha \sqsubset \beta \sqsubset \alpha$.

Conversely, suppose that $\alpha \sqsubset \beta \sqsubset \alpha$. Then, by Eq.(4), $pos_u(\alpha) = pos_u(\beta)$ and $\alpha \neq \beta$. Moreover, by Eq.(4), $\alpha \not\rightleftharpoons \beta$.

L4 : Suppose that $\alpha \neq \beta$ and $\ell(\alpha) = \ell(\beta)$. Then $pos_u(\alpha) \neq pos_u(\beta)$ and so, by Eq.(4), $\alpha \rightleftharpoons \beta$.

As a result, $los \in \mathsf{LOS}$.

Suppose now that $\alpha = \langle a, i \rangle \in \Delta$ and $\beta = \langle a, j \rangle \in \Delta$, where $i \neq j$. Then $i < j \iff pos_u(\alpha) = pos_u(\beta)$. Hence, by Eq.(4), the first part of Eq.(3) holds.

Finally, suppose that $\alpha = \langle a, i\rangle \in \Delta$ and $\beta = \langle b, k\rangle \in \Delta$ are such that $\alpha \sqsubset^{\circledast} \beta$ and $\alpha \neq \beta$. Then, by Eq.(4), $pos_u(\alpha) = pos_u(\beta)$. Hence, by $u \in \mathsf{SSEQ}_\theta$, we have $\langle a, b\rangle \in \mathsf{sim}$, and so the second part of Eq.(3) holds.

As a result, $los \in \mathsf{LOS}_\theta$. □

Labelling the layer decomposition of the order structure of a step sequence in SSEQ_θ is the same as listing this step sequence with explicit action occurrences.

Proposition 18. *If* $u \in \mathsf{SSEQ}_\theta$ *then* $\tau_{\mathsf{sseq2los}(u)} = occseq(u)$.

Proof. Suppose that $occseq(u) = \Delta_1 \ldots \Delta_k$ and $los = \mathsf{sseq2los}(u) = \langle \Delta, \rightleftharpoons, \sqsubset, \ell\rangle$. Clearly, $\Delta_1, \ldots, \Delta_k$ is a partition of Δ. Moreover, from Eq.(4) it follows that

$$\rightleftharpoons \,=\, \bigcup_{i \neq j} \Delta_i \times \Delta_j \qquad \sqsubset \,=\, \bigcup_{i \leq j} \Delta_i \times \Delta_j \setminus id_\Delta \qquad \prec \,=\, \bigcup_{i<j} \Delta_i \times \Delta_j \;.$$

Hence, by Proposition 15(1), $\tau_{\mathsf{sseq2los}(u)} = occseq(u)$. □

Theorem 8. $\mathsf{SSEQ}_\theta \xrightarrow{\mathsf{sseq2los}} \mathsf{LOS}_\theta \xrightarrow{\mathsf{los2sseq}} \mathsf{SSEQ}_\theta$ *are inverse bijections.*

Proof. Suppose that $u \in \mathsf{SSEQ}_\theta$. By Proposition 18, $\tau_{\mathsf{sseq2los}(u)} = occseq(u)$. Hence we obtain $\mathsf{los2sseq}(\mathsf{sseq2los}(u)) = \ell(\tau_{\mathsf{sseq2los}(u)}) = \ell(occseq(u)) = u$. □

Finally, using the bijective correspondence between step sequences and their underlying labelled order structures, we lift the concept of trace equivalence to the level of LO-structures.

Let $los, los' \in \mathsf{LOS}_\theta$. Then $los \mathrel{\dot{\approx}_\theta} los'$ if $\mathsf{los2sseq}(los) \approx_\theta \mathsf{los2sseq}(los')$. We denote by $\dot{\equiv}_\theta$ the reflexive transitive closure of $\dot{\approx}_\theta$, and by TLOS_θ the set of equivalence classes of $\dot{\equiv}_\theta$, called LOS-*traces.*

Theorem 9. $\mathsf{TSSEQ}_\theta \xrightarrow{\mathsf{sseq2los}} \mathsf{TLOS}_\theta \xrightarrow{\mathsf{los2sseq}} \mathsf{TSSEQ}_\theta$ *are inverse bijections.*

Proof. Follows from Theorem 8 and the definition of TLOS_θ. □

The above result provides a strong as well as convenient method for shifting between the language-theoretic (i.e., generalised traces) and order-theoretic (i.e., LOS-traces) descriptions of concurrent histories.

6.2 Step sequences and dependence graphs

The information assembled in the alphabet θ is sufficient to capture the intrinsic dependencies between events involved in a single execution. For this we define the mapping[6]

$$\boxed{\mathsf{SSEQ}_\theta \xrightarrow{\mathsf{sseq2os}} \mathsf{OS} \qquad u \xmapsto{\mathsf{sseq2os}} \langle \Delta, \rightleftharpoons, \sqsubset, \ell\rangle}$$

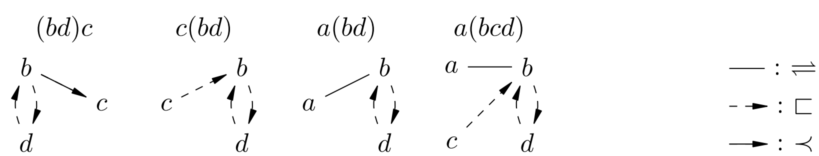
[6] Note that actually θ is a parameter of this mapping as its definition depends on sim and seq. However, as θ is fixed, we omit this reference.

where $\Delta = occ(u)$, for all $\alpha, \beta \in \Delta$ with $pos_u(\alpha) = k$ and $pos_u(\beta) = m$:

$$
\begin{array}{llll}
\alpha \rightleftharpoons \beta \ \textit{if} & \langle \ell(\alpha), \ell(\beta) \rangle \notin \mathsf{sim} \cap \mathsf{seq} & (\in \mathsf{ssi} \cup \mathsf{wdp} \cup \mathsf{rig} \cup \mathsf{inl}) & \wedge\ k < m \\
\qquad\quad \textit{or} & \langle \ell(\alpha), \ell(\beta) \rangle \notin \mathsf{sim} \cap \mathsf{seq}^{-1} & (\in \mathsf{ssi} \cup \mathsf{sse} \cup \mathsf{rig} \cup \mathsf{inl}) & \wedge\ k > m \\
\alpha \sqsubset \beta \ \textit{if} & \langle \ell(\alpha), \ell(\beta) \rangle \notin \mathsf{seq} \cap \mathsf{seq}^{-1} & (\in \mathsf{ssi} \cup \mathsf{sse} \cup \mathsf{wdp} \cup \mathsf{rig}) & \wedge\ k < m \\
\qquad\quad \textit{or} & \langle \ell(\alpha), \ell(\beta) \rangle \in \mathsf{sim} \setminus \mathsf{seq}^{-1} & (\in \mathsf{ssi} \cup \mathsf{sse}) & \wedge\ k = m
\end{array}
\tag{5}
$$

We refer to $\mathsf{sseq2os}(u)$ as the *dependence graph* of u. The definition of dependence graph explicitly indicates if two action occurrences are weakly causally related and/or mutual exclusive or neither based on their relative order in the sequence and their mutual relation as given in θ. Consider, e.g., the first line in the definition: two occurrences, that are not in the same step and have labels that cannot be sequentialised when in the same step, are to be connected by the mutex relation. As another example, the last line states that two occurrences are weakly causally related if they occur in the same step a serialisation with the second action first occurring is not possible. Note that definition given above refers also to the semantical relationships between actions as discussed in Section 4.4. We will come back to that later.

Example 5. Let θ_0 be as in Example 2. The following are example dependence graphs generated from step sequences in SSEQ_{θ_0}:



With the next proposition we establish a number of properties involving dependence graphs: first of all that the mapping $\mathsf{sseq2os}$ is well-defined; in addition using the additional semantical relationships from Section 4.4 all possible relations in a dependence graph can be characterised in a concise way.

Proposition 19. *Let $u \in \mathsf{SSEQ}_\theta$ and $os = \mathsf{sseq2os}(u) = \langle \Delta, \rightleftharpoons, \sqsubset, \ell \rangle$.*

(i) *$\rightleftharpoons$ is symmetric, and both $\rightleftharpoons$ and $\sqsubset$ are irreflexive.*
(ii) *If $\alpha, \beta \in \Delta$ with $pos_u(\alpha) = k$ and $pos_u(\beta) = m$, then:*

$$
\begin{array}{lllll}
\alpha \not\sqsubset \beta \ \wedge\ \beta \not\sqsubset \alpha \ \wedge\ \alpha \not\rightleftharpoons \beta & \Longleftrightarrow & \langle \ell(\alpha), \ell(\beta) \rangle \in \mathsf{con} & & \\
\alpha \sqsubset \beta \ \wedge\ \beta \sqsubset \alpha \ \wedge\ \alpha \not\rightleftharpoons \beta & \Longleftrightarrow & \langle \ell(\alpha), \ell(\beta) \rangle \in \mathsf{ssi} & \wedge & k = m \\
\alpha \not\sqsubset \beta \ \wedge\ \beta \not\sqsubset \alpha \ \wedge\ \alpha \rightleftharpoons \beta & \Longleftrightarrow & \langle \ell(\alpha), \ell(\beta) \rangle \in \mathsf{inl} & \wedge & k \neq m \\
\alpha \sqsubset \beta \ \wedge\ \beta \not\sqsubset \alpha \ \wedge\ \alpha \not\rightleftharpoons \beta & \Longleftrightarrow & \langle \ell(\alpha), \ell(\beta) \rangle \in \mathsf{sse} & \wedge & k \leq m \\
\alpha \sqsubset \beta \ \wedge\ \beta \not\sqsubset \alpha \ \wedge\ \alpha \rightleftharpoons \beta & \Longleftrightarrow & \langle \ell(\alpha), \ell(\beta) \rangle \in \mathsf{ssi} \cup \mathsf{wdp} \cup \mathsf{rig} & \wedge & k < m
\end{array}
$$

(iii) *If $\langle a, i \rangle, \langle a, j \rangle \in \Delta$ then $\langle a, i \rangle \prec \langle a, j \rangle \Longleftrightarrow i < j$.*
(iv) *If $\alpha \sqsubset^{\circledast} \beta$ and $\alpha \neq \beta$, then $pos_u(\alpha) = pos_u(\beta)$ and $\langle \ell(\alpha), \ell(\beta) \rangle \in \mathsf{sim}$.*
(v) $\mathsf{os2loss}(os) \subseteq \mathsf{LOS}_\theta$.

(vi) *The mapping* $\mathsf{SSEQ}_\theta \xrightarrow{\mathsf{sseq2os}} \mathsf{OS}$ *is well-defined.*

Proof. ***(i):*** $\rightleftharpoons$ is symmetric by $(\mathsf{sim} \setminus \mathsf{seq}^{-1})^{-1} = \mathsf{sim}^{-1} \setminus \mathsf{seq} = \mathsf{sim} \setminus \mathsf{seq}$ and Eq.(5). Clearly, it is also irreflexive by Eq.(5). Also, by Eq.(5) and the fact that $\mathsf{sim} \setminus \mathsf{seq}^{-1}$ is irreflexive, $\sqsubset$ is irreflexive.

(ii): Follows directly from Eq.(5).

(iii): Clearly, $\langle a, i\rangle \not\sqsubset \langle a, j\rangle$ for $i = j$, and so without loss of generality we can assume $i < j$ and $pos_u(\langle a, i\rangle) < pos_u(\langle a, j\rangle)$.

Then, by Eq.(5) and $\langle a, a\rangle \notin (\mathsf{sim}\cap\mathsf{seq})\cup(\mathsf{sim}\cap\mathsf{seq}^{-1})$, $\langle a, i\rangle \rightleftharpoons \langle a, j\rangle \rightleftharpoons \langle a, i\rangle$. Moreover, $\langle a, a\rangle \notin \mathsf{seq}\cap\mathsf{seq}^{-1}$ and so, by Eq.(5), $\langle a, i\rangle \sqsubset \langle a, j\rangle$. We then observe that $\langle a, j\rangle \sqsubset \langle a, i\rangle$ is impossible by Eq.(5) and $pos_u(\langle a, i\rangle) < pos_u(\langle a, j\rangle)$.

(iv): By Eq.(5), $\alpha \sqsubset^{\circledast} \beta$ implies $pos_u(\alpha) = pos_u(\beta)$. This and $\alpha \neq \beta$ means that $\ell(\alpha) \neq \ell(\beta)$. Hence $\langle \ell(\alpha), \ell(\beta)\rangle \in \mathsf{sim}$ since $u \in \mathsf{SSEQ}_\theta$.

(v): The first part of Eq.(3) follows from (iii), and the second from (iv).

(vi): We need to show that os is label-linear and separable. The former follows from (iii). Moreover, if $\alpha \sqsubset^{\circledast} \beta$ and $\alpha \neq \beta$ then, by (iv), $pos_u(\alpha) = pos_u(\beta)$. Hence, by Eq.(5), $\alpha \not\rightleftharpoons \beta$ and so os is separable. $\square$

Summarising, we now have two kinds of order structures to capture the intrinsic dependencies of action occurrences in the step sequences of SSEQ_θ, namely dependence graphs and their closure in the form of generalised mutex order structures (invariant order structures):

$$\boxed{\mathsf{OS}_\theta = \mathsf{sseq2os}(\mathsf{SSEQ}_\theta)} \quad \text{and} \quad \boxed{\mathsf{GMOS}_\theta = \mathsf{os2gmos}(\mathsf{OS}_\theta)}$$

6.3 Dependence graphs and traces

We will now investigate the relations between dependence graphs, step sequences, and traces. First of all, every step sequence belongs through its underlying order structure, is one of the observations of the history associated with its dependence graph.

Proposition 20. *For every* $u \in \mathsf{SSEQ}_\theta$*,* $u \in \mathsf{los2sseq} \circ \mathsf{os2loss} \circ \mathsf{sseq2os}(u)$.

Proof. Let $os = \mathsf{sseq2os}(u) = \langle \Delta, \rightleftharpoons, \sqsubset, \ell\rangle$. By Theorem 8, it suffices to show that $los = \mathsf{sseq2los}(u)$ belongs to $\mathsf{os2loss}(os)$.

Thus we prove that los is a saturated version of os. Suppose that $\alpha \rightleftharpoons \beta$. Then, by Eq.(5), $pos_u(\alpha) \neq pos_u(\beta)$. Hence $\alpha \rightleftharpoons_{los} \beta$. Next suppose that $\alpha \sqsubset \beta$. Then, by Eq.(5), $pos_u(\alpha) \leq pos_u(\beta)$. Hence $\alpha \sqsubset_{los} \beta$. As a result, $los \in \mathsf{os2loss}(os)$. $\square$

The second results states that equivalent step sequences generate the same dependence graphs.

Proposition 21. *For all* $u, w \in \mathsf{SSEQ}_\theta$*,* $u \equiv_\theta w$ *implies* $\mathsf{sseq2os}(u) = \mathsf{sseq2os}(w)$.

Proof. Let $\mathsf{sseq2os}(u) = \langle \Delta, \rightleftharpoons, \sqsubset, \ell \rangle$ and $\mathsf{sseq2os}(w) = \langle \Delta, \rightleftharpoons', \sqsubset', \ell \rangle$. It suffices to show the result in the following two cases.
Case 1: $u = AB$, $w = BA$ and $A \times B \subseteq \mathsf{seq} \cap \mathsf{seq}^{-1}$. Then, by seq being irreflexive, we have that $occseq(u) = \Delta_1 \Delta_2$ and $occseq(w) = \Delta_2 \Delta_1$, for some Δ_1 and Δ_2. Clearly, $\rightleftharpoons = \rightleftharpoons'$ as $(\mathsf{sim} \cap \mathsf{seq})^{-1} = \mathsf{sim} \cap \mathsf{seq}^{-1}$. Moreover, $\sqsubset = \sqsubset'$ as the following holds, by Eq.(5) and $A \times B \subseteq \mathsf{seq} \cap \mathsf{seq}^{-1}$:

$$((\Delta_1 \times \Delta_2) \cup (\Delta_2 \times \Delta_1)) \cap \sqsubset = ((\Delta_1 \times \Delta_2) \cup (\Delta_2 \times \Delta_1)) \cap \sqsubset' = \varnothing\ .$$

Case 2: $u = AB$, $w = A \cup B$ and $A \times B \subseteq \mathsf{seq}$. Then, by seq being irreflexive, we have that $occseq(u) = \Delta_1 \Delta_2$ and $occseq(w) = \Delta_1 \uplus \Delta_2$, for some Δ_1 and Δ_2. We then have $\rightleftharpoons = \rightleftharpoons' = \varnothing$ as $A \times B \subseteq \mathsf{sim} \cap \mathsf{seq}$.

Suppose that $\alpha \in \Delta_1$ and $\beta \in \Delta_2$. Then $\alpha \sqsubset \beta$ iff $\langle \ell(\alpha), \ell(\beta) \rangle \in \mathsf{sim} \setminus \mathsf{seq}^{-1}$. Moreover, $\alpha \sqsubset' \beta$ iff $\langle \ell(\alpha), \ell(\beta) \rangle \notin \mathsf{seq} \cap \mathsf{seq}^{-1}$ iff $\langle \ell(\alpha), \ell(\beta) \rangle \in \mathsf{sim} \setminus \mathsf{seq}^{-1}$ (since $\langle \ell(\alpha), \ell(\beta) \rangle \in \mathsf{sim} \cap \mathsf{seq}$).
Suppose now that $\alpha \in \Delta_2$ and $\beta \in \Delta_1$, and so $\langle \ell(\beta), \ell(\alpha) \rangle \in \mathsf{seq}$. Then $\alpha \not\sqsubset' \beta$, by Eq.(5). If $\alpha \sqsubset \beta$ then $\langle \ell(\alpha), \ell(\beta) \rangle \in \mathsf{sim} \setminus \mathsf{seq}^{-1}$, contradicting $\langle \ell(\beta), \ell(\alpha) \rangle \in \mathsf{seq}$. As a result, $\sqsubset = \sqsubset'$. □

Consequently we can associate a single dependence graph with a generalised trace in the following way:

$$\boxed{\mathsf{TSSEQ}_\theta \xrightarrow{\mathsf{sseq2os}} \mathsf{OS} \qquad [\![u]\!]_\theta \xmapsto{\mathsf{sseq2os}} \mathsf{sseq2os}(u)}$$

Proposition 22. *The mapping* $\mathsf{TSSEQ}_\theta \xrightarrow{\mathsf{sseq2os}} \mathsf{OS}$ *is well-defined.*

Proof. Follows from Proposition 21. □

Now we turn to the reverse question whether all step sequences defined by a single dependence graph are equivalent (and thus form a generalised trace). To deal with this it is convenient to introduce the notion of a step which cannot be split.

A *min-step* is a nonempty set of actions $A \in \mathbb{S}_\theta$ such that there are no steps B, C satisfying $A = B \uplus C$ and $B \times C \subseteq \mathsf{seq}$. A step sequence $u \in \mathsf{SSEQ}_\theta$ is *thin* if it is composed of min-steps. The family of all such step sequences will be denoted by $\mathsf{SSEQ}_\theta^{thin}$.

Example 6. Let θ_0 be as in Example 2. The generalised trace $[\![a(bcd)]\!]_{\theta_0}$ contains three thin step sequences: $ac(bd)$, $ca(bd)$ and $c(bd)a$; and three non-thin ones: $a(bcd)$, $(bcd)a$ and $(ac)(bd)$. ◇

Any step sequence can be 'flattened' to yield an equivalent thin step sequence.

Proposition 23. *For every* $u \in \mathsf{SSEQ}_\theta$ *there is* $w \in \mathsf{SSEQ}_\theta^{thin}$ *such that* $u \equiv_\theta w$.

Proof. Let $w = A_1 \dots A_k$ be a longest step sequence such that $u \equiv_\theta w$. Suppose that w is not thin, and A_i is not a min-step, for some $i \le k$. This means that there are steps B, C such that $A_i = B \uplus C$ and $B \times C \subseteq \mathsf{seq} \cap \mathsf{sim}$. Hence $w \approx_\theta A_1 \dots A_{i-1} B C A_{i+1} \dots A_k$, contradicting the choice of w. □

Finally we are ready for the basic result that we need for our proof of the equivalence of all step sequences defined by the dependence graph of a trace. The proof relies on several auxiliary observations, formulated as internal Lemmata. We start from a thin step sequence and its dependence graph. With the min-steps as 'atomic' building blocks, we first follow more or less the classical approach for Mazurkiewicz traces and their dependence graphs in which the atoms are singleton sets. Recalling that a dependence graph collects all causal (necessary) relations between the min-steps with all other relations being observational and specific to the initial step sequence, we are free to change the order of min-steps as long as we do not violate the invariant causality of the dependence graph. The result is (another) linearisation (cf. Section 3 and Proposition 1), that is equivalent with the given step sequence. This can be repeated and finally we also combine min-steps into larger steps, still obeying the restrictions of causality imposed by the dependence graph that guarantees equivalence of the thus obtained new step sequence. Note that in the statement and proof, we do not use step sequences to establish the equivalence, but rather their order-theoretic counterparts.

Proposition 24. *Let* $u \in \mathsf{SSEQ}_\theta^{thin}$ *and* $os = \mathsf{sseq2os}(u) = \langle \Delta, \rightleftharpoons, \sqsubset, \ell \rangle$. *Then* $\mathsf{sseq2los}(u) \doteqdot_\theta los$, *for every* $los \in \mathsf{os2loss}(os)$.

Proof. Let $u = A_1 \dots A_k$ and $occseq(u) = \Delta_1 \dots \Delta_k$. Moreover, let $\mathcal{X}$ be the set of all equivalence classes of $\sqsubset^{\circledast}$, and $\ll$ be a binary relation over $\mathcal{X}$ such that $X \ll Y$ if $(X \times Y) \cap \rightleftharpoons \neq \varnothing$ and $(X \times Y) \cap \sqsubset \neq \varnothing$.

Lemma 1. $\ll$ *is an acyclic relation, and* $occseq(u)$ *is a linearisation of* $\ll$.

Proof. The first part is obvious, and the second follows from Eq.(4). □

Lemma 2. $(X \times X) \cap \rightleftharpoons = \varnothing$, *for every* $X \in \mathcal{X}$.

Proof. Follows from os being an order structure (and its separability). □

Lemma 3. *If* ξ *is a linearisation of* $\ll$ *then* $los_\xi = \mathsf{sseq2los}(\ell(\xi)) \in \mathsf{os2loss}(os)$ *and* $\tau_{los_\xi} = \xi$.

Proof. Follows from Lemmata 1 and 2. □

Lemma 4. $\mathcal{X} = \{\Delta_1, \dots, \Delta_k\}$.

Proof. Consider A_i and Δ_i. Since A_i is a min-step, the graph of the relation $(A_i \times A_i) \setminus \mathsf{seq}$ over A_i is strongly connected. Hence, by Eq.(5), the graph of $\sqsubset$ restricted to the nodes of Δ_i is also strongly connected. Suppose that $\alpha \in \Delta \setminus \Delta_i$ and $\beta \in \Delta_i$ are such that $\alpha \sqsubset^{\circledast} \beta$. Then, by Eq.(5), $pos_u(\alpha) = pos_u(\beta)$ and so $\alpha \in \Delta_i$, a contradiction. It therefore follows that $\Delta_i \in \mathcal{X}$.

The result follows as both $\Delta_1, \dots, \Delta_k$ and $\mathcal{X}$ are partitions of Δ. □

Lemma 5. *Let* $los' \in \mathsf{os2loss}(os)$, $\tau_{los'} = \Phi_1 \dots \Phi_m$ *and* $j \leq m$.

(i) Φ_j *is the union of some* Δ_i*'s.*

(ii) $(\Phi_j \times \Phi_j) \cap \rightleftharpoons = \varnothing$.

(iii) *If there is no* $los'' \in \mathsf{os2loss}(os)$ *such that* $los' \dot{\approx}_\theta los''$ *and the length of* $\tau_{los''}$ *is greater than* m*, then* $\tau_{los'}$ *is a linearisation of* $\ll$.

(iv) *If* X *and* Y *are distinct elements of* $\mathcal{X}$ *satisfying* $X \not\ll Y$ *and* $Y \not\ll X$*, then* $\ell(X) \times \ell(Y) \subseteq \mathsf{seq} \cap \mathsf{seq}^{-1}$.

(v) *If* ξ *is a linearisation of* $\ll$ *then* $los_\xi \dot{\equiv}_\theta los'$.

Proof. ***(i):*** Follows from the fact that if $\Phi_j \cap \Delta_i \neq \varnothing$ then $\Delta_i \subseteq \Phi_j$ as Δ_i is an equivalence class of $\sqsubset^{\circledast}$.

(ii): Follows from Proposition 16.

(iii): By part (i), $\Phi_j = \Delta_{i_1} \uplus \cdots \uplus \Delta_{i_l}$. Suppose that $l > 1$. Since $\ll$ is acyclic, there is $s \leq l$ such that there is no $z \in Z = \{i_1, \ldots, i_l\} \setminus \{i_s\}$ with $\Delta_z \ll \Delta_{i_s}$ (i.e., Δ_{i_s} is $\ll$-minimal).

Consider next the nonempty sets Δ_{i_s} and $\Phi_j \setminus \Delta_{i_s}$. Suppose $\alpha \in \Delta_{i_s}$ and $\beta \in \Delta_z$, for some $z \in Z$, which means that $pos_u(\alpha) \neq pos_u(\beta)$. By Eq.(5) and $\alpha \not\rightleftharpoons_{los'} \beta$, we have $\langle \ell(\alpha), \ell(\beta) \rangle \in \mathsf{sim}$. Suppose that $\langle \ell(\alpha), \ell(\beta) \rangle \notin \mathsf{seq}$. Then, by Eq.(5), $\beta \sqsubset \alpha$, contradicting the choice of Δ_{i_s} ($\ll$-minimality). As a result, $A_{i_s} \times \bigcup_{z \in Z} A_z \subseteq \mathsf{seq}$. Hence

$$los'' = \mathsf{sseq2los}(\ell(\Phi_1 \ldots \Phi_{i_s - 1} \Delta_{i_s} (\bigcup_{z \in Z} \Delta_z) \Phi_{i_s + 1} \ldots \Phi_m))$$

is such that $los' \dot{\approx} los''$ and $los'' \in \mathsf{os2loss}(os)$. This produces a contradiction with the choice of los'. Hence $\tau_{los'}$ is a linearisation of $\ll$.

(iv): Let $\alpha \in X$ and $\beta \in Y$. Then, by Proposition 19(ii) (1st or 3rd line), we have $\langle \ell(\alpha), \ell(\beta) \rangle \in (\mathsf{sim} \cap \mathsf{seq} \cap \mathsf{seq}^{-1}) \cup (\mathsf{seq} \cap \mathsf{seq}^{-1} \setminus \mathsf{sim}) = \mathsf{seq} \cap \mathsf{seq}^{-1}$.

(v): By (iii), we can assume that that $\tau_{los'}$ is a linearisation of $\ll$. By Proposition 1, there are linearisations $v_1, \ldots, v_r$ of $\ll$ such that $\tau_{los'} = v_1 \sim \cdots \sim v_k = \xi$. We then observe that the result follows from (iv). □

We now observe that the proposition follows from Lemmata 1 and 5. □

We can now conclude that all step sequences generated from a dependence graph are equivalent.

Proposition 25. *If* $u \in \mathsf{SSEQ}_\theta$ *and* $w \in \mathsf{los2sseq} \circ \mathsf{os2loss} \circ \mathsf{sseq2os}(u)$*, then* $u \equiv_\theta w$.

Proof. Follows from Propositions 21, 23 and 24. □

From the results on generalised traces, on order structures, and their interconnections we can now conclude that we have achieved the main aim of this paper.

Theorem 10 (equivalent behaviour representations). *The diagram in Figure 5 commutes.*

Proof. Follows from Theorems 7 and 9, Proposition 22, and the following argument.

Let $u \in \mathsf{SSEQ}_\theta$. Suppose that $w \equiv_\theta u$. Then, by Proposition 21, $\mathsf{sseq2os}(w) = \mathsf{sseq2os}(u)$. Hence, by Proposition 20, $w \in \mathsf{los2sseq} \circ \mathsf{os2loss} \circ \mathsf{sseq2os}(u)$. As a result, $[\![u]\!]_\theta \subseteq \mathsf{los2sseq} \circ \mathsf{os2loss} \circ \mathsf{sseq2os}(u)$. Moreover, by Proposition 25, $\mathsf{los2sseq} \circ \mathsf{os2loss} \circ \mathsf{sseq2os}(u) \subseteq [\![u]\!]_\theta$. Hence $[\![u]\!]_\theta = \mathsf{los2sseq} \circ \mathsf{os2loss} \circ \mathsf{sseq2os}(u)$. □

We have therefore obtained a counterpart of the schematic behaviour diagram of Figure 1, providing details of all the domains and mappings involved. In addition, the diagram in Figure 5 provides one more domain, TLOS_θ, which provides a technically convenient bridge between the language-theoretic domain of generalised traces and the order-theoretic domain of invariant order structures. In Figure 1 — and indeed the standard approach of Mazurkiewicz traces — such a bridge is established 'on-the-fly' by an implicit identification of a sequence of actions with the corresponding labelled total order.

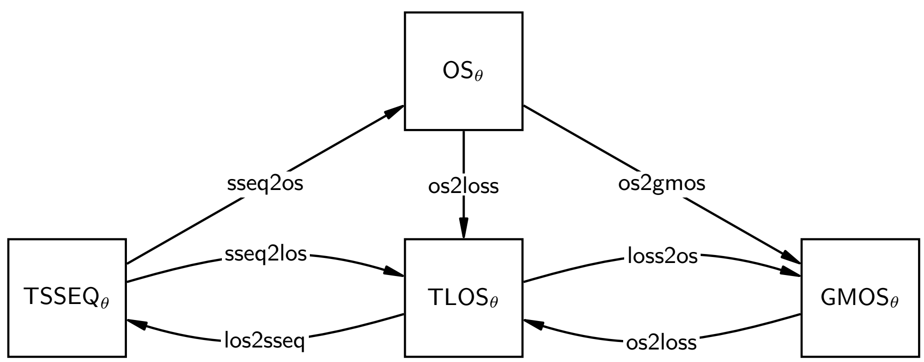
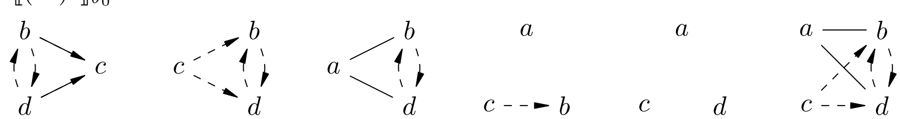


Fig. 5. Behaviour diagram for generalised traces and the underlying order structures.

Example 7. Let θ_0 be as in Example 2. The following are example invariant order structures (i.e., GMO-structures) corresponding to generalised traces in $\mathsf{TSSEQ}_{\theta_0}$:

$[\![(bd)c]\!]_{\theta_0}$ $[\![c(bd)]\!]_{\theta_0}$ $[\![a(bd)]\!]_{\theta_0}$ $[\![a(bc)]\!]_{\theta_0}$ $[\![a(cd)]\!]_{\theta_0}$ $[\![a(bcd)]\!]_{\theta_0}$



7 Conclusions

In this paper, we discuss how to extend Mazurkiewicz traces taking step sequences as smallest units of observation rather than single action. We aimed at staying close to original trace philosophy — embodied by the schematic commutative diagram of Figure 1 — making as few as possible, and as light as possible, design choices along the way. Such an approach has led us to to a general set-up of fundamental concurrency alphabets allowing intuitive classifications fitting established, and as yet untried, trace models. Then, for technical convenience, we switched to an equivalent presentation of trace model, captured using generalised concurrency alphabets.

The main aim of this paper was then to develop relational structures matching extended traces and step sequences in the same way as partial orders match Mazurkiewicz traces and total orders match sequences of action occurrences. The result of our investigation is captured by the commuting diagram of Figure 5. In essence, it shows that the generalised traces as proposed in this paper are actually the most general version of Mazurkiewicz traces for step sequences. We end our discussion looking at the expressiveness of generalised traces.

So far we have demonstrated that generalised traces can be represented by GMOS-structures. A question might therefore arise as to whether such structures are really necessary, or perhaps a class of simpler order structures would suffice. We will now show that it this is not the case.

Proposition 26. *Let os be an order structure with an injective labelling. Then there is a generalised concurrency alphabet θ and a step sequence $u \in \mathsf{SSEQ}_\theta$ such that os is isomorphic to* $\mathsf{sseq2os}(u)$.

Proof. Let $os = \langle \Delta, \rightleftharpoons, \sqsubset, \ell \rangle$. Since ℓ is injective, we may assume that each $\alpha \in \Delta$ is of the form $\langle a, 1 \rangle$ with $\ell(\alpha) = a$. Hence Δ is an event domain.

By Proposition 14, there is $los \in \mathsf{os2loss}(os) \neq \varnothing$. Clearly, $los \in \mathsf{LOS}_\theta$, for any generalised concurrency alphabet $\theta = \langle \ell(\Delta), \mathsf{sim}, \mathsf{seq} \rangle$. We will now show how to construct sim and seq in order to obtain a desired alphabet.

First, we observe that the layer sequence τ_{los} is well-defined even though θ is not fully defined. Moreover, τ_{los} can be treated as a step sequence over the alphabet Δ. We then construct sim and seq as follows, by taking all pairs of distinct $\alpha, \beta \in \Delta$ with $k = pos_{\tau_{los}}(\alpha)$ and $m = pos_{\tau_{los}}(\beta)$:

Case 1: $\alpha \not\sqsubset \beta \wedge \beta \not\sqsubset \alpha \wedge \alpha \not\rightleftharpoons \beta$. Then we add $\langle \ell(\alpha), \ell(\beta) \rangle$ and $\langle \ell(\beta), \ell(\alpha) \rangle$ to both sim and seq.

Case 2: $\alpha \sqsubset \beta \wedge \beta \sqsubset \alpha \wedge \alpha \not\rightleftharpoons \beta$. Then $k = m$ and we add $\langle \ell(\alpha), \ell(\beta) \rangle$ and $\langle \ell(\beta), \ell(\alpha) \rangle$ to sim.

Case 3: $\alpha \not\sqsubset \beta \wedge \beta \not\sqsubset \alpha \wedge \alpha \rightleftharpoons \beta$. Then $k \neq m$ and we add $\langle \ell(\alpha), \ell(\beta) \rangle$ and $\langle \ell(\beta), \ell(\alpha) \rangle$ to seq.

Case 4: $\alpha \sqsubset \beta \wedge \beta \not\sqsubset \alpha \wedge \alpha \not\rightleftharpoons \beta$. Then $k \leq m$ and we add $\langle \ell(\alpha), \ell(\beta) \rangle$ and $\langle \ell(\beta), \ell(\alpha) \rangle$ to sim, and $\langle \ell(\alpha), \ell(\beta) \rangle$ to seq.

Case 5: $\alpha \sqsubset \beta \wedge \beta \not\sqsubset \alpha \wedge \alpha \rightleftharpoons \beta$. Then $k < m$ and we do not add anything.

Note that the above construction follows from the characterisation provided by Proposition 19(2).

We observe that θ is a generalised concurrency alphabet. Indeed, sim and seq are irreflexive by construction and the fact that $\alpha \neq \beta$ implies $\ell(\alpha) \neq \ell(\beta)$. Moreover, sim is symmetric by construction, and $\mathsf{seq} \setminus \mathsf{sim}$ is symmetric because it can only acquire pairs of elements in Case 3.

Let $u = \mathsf{los2sseq}(los) = \ell(\tau_{los})$. Then $u \in \mathsf{SSEQ}_\theta$ follows from the fact that, in the above construction, if $k = m$ then we have Case 1 or Case 2 or Case 4. We then observe that $os = \mathsf{sseq2os}(u)$ follows from Eq.(5). □

Although in the above we assumed injective labelling, the result we obtained demonstrates that generalised concurrency alphabets can generate all the complex patterns involving causal relationships captured by GMO-structures. We therefore the full power of such structures. In fact, the last demonstrates somewhat stronger property, namely that any order structure can provide a skeleton for the dependence graph of some generalised trace.

Finally, there are concurrent histories / invariant order structures that cannot be generated by any generalised concurrency alphabet. Consider, for example, the GMO-structure *gmos* (similar to ios_0 in Example 4) with three action occurrences, $\langle a, 1\rangle$, $\langle a, 2\rangle$ and $\langle b, 1\rangle$, and two relationships $\langle a, 1\rangle \prec \langle a, 2\rangle$ and $\langle a, 1\rangle \prec \langle b, 1\rangle$ (the corresponding history *hloss* contains three LO-structures, los_i $(i = 1, 2, 3)$, such that $\tau_{los_1} = aab$, $\tau_{los_2} = aba$, and $\tau_{los_3} = a(ab)$). One can see that there is no generalised concurrency alphabet θ such that $\{los_1, los_2, los_3\} \in \mathsf{TLOS}_\theta$ and $gmos \in \mathsf{GMOS}_\theta$. The intuitive reason is that the first occurrence of a causes b to occur, so a and b are dependent, but the second occurrence of a is concurrent with b, and so a and b are independent. However, in any alphabet θ the relationship between a and b is *static* and cannot depend on a specific occurrence of a.

Acknowledgments

This research was supported by the Polish National Science Center under the grant No.2013/09/D/ST6/03928, and the EPSRC GAELS and UNCOVER projects.

References

1. Diekert, V., Rozenberg, G. (eds.): The Book of Traces. River Edge, NJ, USA (1995)
2. Gaifman, H., Pratt, V.R.: Partial order models of concurrency and the computation of functions. In: LICS, pp. 72–85. IEEE Computer Society (1987)
3. Hoogeboom, H.J., Rozenberg, G.: Dependence graphs. In: V. Diekert, G. Rozenberg (eds.) The Book of Traces, pp. 43–67. World Scientific (1995)
4. Janicki, R., Kleijn, J., Koutny, M., Mikulski, L.: Causal structures for general concurrent behaviours. In: M.S. Szczuka, L. Czaja, M. Kacprzak (eds.) CS&P, *CEUR Workshop Proceedings*, vol. 1032, pp. 193–205. CEUR-WS.org (2013)
5. Janicki, R., Koutny, M.: Invariants and paradigms of concurrency theory. In: E.H.L. Aarts, J. van Leeuwen, M. Rem (eds.) PARLE (2), *Lecture Notes in Computer Science*, vol. 506, pp. 59–74. Springer (1991)
6. Janicki, R., Koutny, M.: Structure of concurrency. Theor. Comput. Sci. **112**(1), 5–52 (1993)
7. Janicki, R., Koutny, M.: Semantics of inhibitor nets. Inf. Comput. **123**(1), 1–16 (1995)
8. Janicki, R., Le, D.T.M.: Modelling concurrency with comtraces and generalized comtraces. Inf. Comput. **209**(11), 1355–1389 (2011)

9. Janicki, R., Yin, X., Zubkova, N.: Modeling interval order structures with partially commutative monoids. In: M. Koutny, I. Ulidowski (eds.) CONCUR, *Lecture Notes in Computer Science*, vol. 7454, pp. 425–439. Springer (2012)
10. Juhás, G., Lorenz, R., Mauser, S.: Causal semantics of algebraic petri nets distinguishing concurrency and synchronicity. Fundam. Inform. **86**(3), 255–298 (2008)
11. Juhás, G., Lorenz, R., Neumair, C.: Synthesis of controlled behavior with modules of signal nets. In: J. Cortadella, W. Reisig (eds.) ICATPN, *Lecture Notes in Computer Science*, vol. 3099, pp. 238–257. Springer (2004)
12. Kleijn, J., Koutny, M.: Formal languages and concurrent behaviours. In: G.B. Enguix, M.D. Jiménez-López, C. Martín-Vide (eds.) New Developments in Formal Languages and Applications, *Studies in Computational Intelligence*, vol. 113, pp. 125–182. Springer (2008)
13. Kleijn, J., Koutny, M.: Mutex causality in processes and traces of general elementary nets. Fundam. Inform. **122**(1-2), 119–146 (2013)
14. Le, D.T.M.: On three alternative characterizations of combined traces. Fundam. Inform. **113**(3-4), 265–293 (2011)
15. Mazurkiewicz, A.: Concurrent program schemes and their interpretations. DAIMI Rep. PB 78, Aarhus University (1977)
16. Mazurkiewicz, A.W.: Basic notions of trace theory. In: J.W. de Bakker, W.P. de Roever, G. Rozenberg (eds.) REX Workshop, *Lecture Notes in Computer Science*, vol. 354, pp. 285–363. Springer (1988)
17. Mikulski, L.: Algebraic structure of combined traces. Logical Methods in Computer Science **9**(3) (2013)
18. Mikulski, L., Koutny, M.: Folded hasse diagrams of combined traces. Inf. Process. Lett. **114**(4), 208–216 (2014)
19. Pratt, V.: Modeling concurrency with partial orders. International Journal of Parallel Programming **15**(1), 33–71 (1986)
20. Szpilrajn, E.: Sur l'extension de l'ordre partiel. Fundam. Math. **16**, 386–389 (1930)
21. Vogler, W.: Partial order semantics and read arcs. Theoretical Computer Science **286**(1), 33–63 (2002)